Metrology of High Aspect Ratio MEMS

A Thesis
Presented to
The Academic Faculty

By

James Franklin Nichols

In Partial Fulfillment
of the Requirements for the Degree

Doctor of Philosophy
In
Mechanical Engineering

Georgia Institute of Technology
May 2004

Metrology of High Aspect Ratio MEMS

Committee Members:
Thomas R. Kurfess, Chair
Paul M. Griffin
Craig C. Henderson
David W. Rosen
F. Levent Degertekin
Date Approved: April 8, 2004

ACKNOWLEDGEMENT

The author would like to first thank his advisor, Dr. Thomas Kurfess, for the invaluable suggestions made over the course of this research. The reading committee's comments and suggestions were also a valuable resource in completing this research. Additionally, the author would like to thank the students of the Precision Machining Research Consortium for their contributions over the years. Lastly, the author would like to thank his parents, Jim and Kalaya Nichols, and Olivia for their unwavering moral support.

This work was partially funded by Sandia National Laboratories and the National Science Foundation under Grant Number DMI-9988664. The government has certain rights in this material. Any opinions, findings and conclusions or recommendations are those of the authors and do not necessarily reflect the views of Sandia National Laboratories or the National Science Foundation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The current tools for geometric analysis of micro-electromechanical systems (MEMS) are primarily limited to those of the semiconductor industry. These tools are suited for measuring entities that are two-dimensional in nature such as lines, circles, and planes. Hardware that is capable of collecting three-dimensional data is typically limited by the slope variations in the surfaces of the part, and cannot accurately capture information from steep sidewalls, particularly in parts fabricated using the LIGA micro-fabrication process.

This research develops a methodology to qualify MEMS, by implementing a novel computer-aided inspection (CAI) software framework. This software platform uses data acquired from current MEMS inspection hardware, and applies newly developed analysis algorithms to geometrically characterize a part. This work implements algorithms for all the procedures typical to a CAI program (e.g., point-to-entity assignment, registration, and data analysis) in addition to new techniques suited for inspection of high aspect ratio MEMS. This methodology describes possible registration errors based on the type of geometries being analyzed and the type of data acquired. Analyses of multiple point clouds with the use of fiducial information are shown to provide a critical link between single point cloud analyses that has heretofore been unrealized.

CHAPTER I

INTRODUCTION

High Aspect Ratio MEMS

Micro-fabrication is a relatively new research field dating back to the late 1960s. Taking advantage of technology used in the integrated circuit (IC) industry, researchers have developed methods to micromachine miniaturized devices which could potentially displace or improve upon their macroscopic counterparts. Since the 1980s, micro-electromechanical systems (MEMS) have been introduced to the mass market in various applications such as pressure sensors, accelerometers, micro-valves, and ink jet nozzle heads. The wide diversity of MEMS applications has led to the rapid growth of MEMS research programs in universities, government agencies, and industry.

Technological advances that are specific to MEMS devices have enabled researchers to fabricate devices with ever increasing aspect ratios (defined as the ratio between the structural height and the minimum lateral dimension of the part). Currently, the LIGA process is seen as the best way to fabricate devices with very tall, parallel sidewalls (up to a few mm) such as those shown in Figure 1.

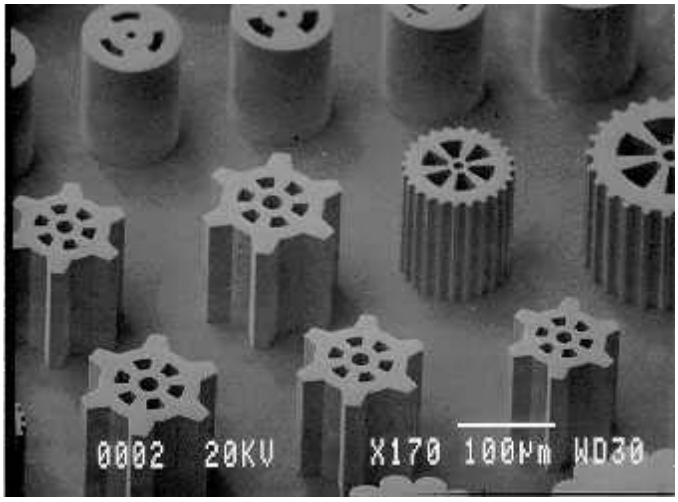

Figure 1: LIGA manufactured parts

As researchers develop more complex three-dimensional shapes, metrology of these parts becomes more difficult. Traditional metrology techniques for IC parts become inadequate, limited by their two-dimensional capabilities.

MEMS Metrology

The current tools for geometric analysis of MEMS are primarily limited to those of the semiconductor industry. These tools are suited for measuring entities that are two-dimensional in nature such as lines, circles, and planes. Some of these measuring devices have been used to analyze MEMS with limited success. Tools that are typically used are traditional optical microscopes, scanning electron microscopes (SEM), white light interferometers, and atomic force microscopes. SEMs and optical microscopes have the appropriate resolution to analyze high aspect ratio MEMS, but are inherently two-dimensional thus making it difficult to extract any useful three-dimensional data from the part. Atomic force microscopy and white light interferometry add the third measurable

dimension; however, these tools are limited by the slope variations in the surfaces of the part. As new micromachining techniques for MEMS devices are developed and allow for truly three-dimensional fabricated parts, dimensional metrology techniques will also have to advance to incorporate full three-dimensional data to accurately qualify these devices.

Coordinate Metrology

In the macroscopic world, coordinate metrology is a mature research field. Many standards exist which detail how to correctly qualify parts and machines. For example, the ASME B89 series of standards covers a wide range of metrology topics including measurement uncertainty (B89.7.3.1), dimensional measurement planning (B89.7.2), performance evaluation of coordinate measuring machines (B89.4.1), and coordinate measuring system software (B89.4.10). The ASME Y14 series covers other metrology aspects such as how to properly apply dimensions and tolerances to part drawings (Y14.5M) and how to properly use graphic symbols in diagrams (Y14.40.1).

For macroscale parts, it is generally accepted that there are two major sub-tasks to the inspection process. These tasks, namely, are data acquisition and data analysis. The intention of separating the tasks of data acquisition and data analysis is to create a more flexible platform for part metrology. As long as the output from the data acquisition side is consistent to the requirements for the input to the data analysis side, these two tasks can be dealt with independently of each other.

Data Acquisition

For parts manufactured by traditional machining methods (e.g., turning, milling, grinding), the coordinate measuring machine (CMM) has become the de facto standard for full part verification. A plethora of research has conducted regarding both calibration and inspection on CMMs. Laser triangulation methods have also become popular techniques to gather copious amounts of data at high speeds, even though the accuracy of the data is not on the same order as with CMMs. Other data acquisition methods commonly used for macroscale parts are optical and interferometric techniques. Regardless of the technique, the typical output of each of these acquisition methods is a list of two-dimensional or three-dimensional points which represent points corresponding to the surfaces of the part.

Data Analysis

Once data acquisition has been completed, the raw data points can be processed to provide useful feedback about the physical part. The two most common analyses are point cloud-to-CAD comparisons and geometric parameter fitting.

In the first type of analysis, the point cloud gathered during the data acquisition phase is loaded into the model space of the ideal CAD model. The primary problem to be solved is the registration of the data to the CAD model. Since the data will typically be in an arbitrary coordinate system with respect to the CAD model coordinate system, these two coordinate systems must be co-located. Once the point cloud is registered to the CAD model, geometric deviations and other statistical measures can be calculated to verify the manufactured part.

In geometric parameter fitting, no CAD model is needed to perform the analysis. In this type of analysis, data subsets of the point cloud are selected and the intended geometry is best-fit to the data. Also, multiple fits can be further analyzed to provide more feedback with regards to multiple geometries (e.g., center distance between two circle fits to two different data sets).

Problem Statement

As previously mentioned, current data acquisition hardware for MEMS verification is limited to the tools of the semiconductor industry. These tools are well-suited to analyzing geometries that are two-dimensional in nature such as line widths. Currently, no hardware exists that can fully characterize geometries of high aspect ratio MEMS. Research is currently underway which will try to address the shortcomings of existing metrology hardware for use in the MEMS industry, but this work is still in its infancy and will not be available for some time. The current inspection procedure for high aspect ratio MEMS involves the best-fit analysis of individual surfaces of a part. No information is currently obtained regarding the overall geometry of the part. This research investigates data analysis techniques that can better qualify high aspect ratio MEMS using current two-dimensional data acquisition techniques.

Objectives of Research

The objective of this research is to develop a methodology to qualify high aspect ratio MEMS, specifically those fabricated using the LIGA process. This research proposes that using current two-dimensional inspection techniques, three-dimensional geometric characteristics can be inferred from parts using multiple two-dimensional scans. Since LIGA parts are technically 2.5-D (i.e. an extruded two-dimensional profile), assumptions about the sidewalls might be made so that only the "tops" and "bottoms" of the parts have to be analyzed to infer 3-D characteristics.

The first objective of this research will be to survey the current state-of-the-art metrology hardware and software for MEMS inspection. Current hardware and software tools are limited, and a survey of current techniques will be described along with their advantages and limitations. After this initial study, the need to use three-dimensional data for inspection of high aspect ratio parts will be evaluated. One of the main attributes of the LIGA process is the ability to fabricate relatively tall, parallel sidewalls. It may be possible to analyze two-dimensional data sets and infer sidewall characteristics with some accuracy.

The next objective of this research is to develop a framework for data analysis that addresses some of the current limitations in metrology software for high aspect ratio MEMS. The foundation of the work will be based on traditional computer-aided inspection theory, and, in addition, new techniques will be developed to address specific requirements for MEMS inspection. This technique, namely, is to incorporate multiple two-dimensional point clouds of data in order to infer three-dimensional characteristics of a part. This work proposes that fiducial information can be incorporated into 2-D scans of

LIGA parts, and that these fiducials can be used in an analytical method to produce quantitative information about the overall part geometry. This methodology will describe the possible registration errors based on the type of geometries being analyzed and the type of data acquired. If multiple two-dimensional data sets can be correctly registered, implementing a methodology to inspect these parts from just the planar surfaces (i.e., top and bottom) will provide new information about the overall geometry of high aspect ratio MEMS.

Once a methodology is established, development of a software platform to implement the proposed methodology will be investigated. This work will implement algorithms for all the procedures typical to a computer-aided inspection program (e.g., point-to-entity assignment, registration, and data analysis), but the main contribution of this work will be the development of new algorithms specifically derived for three-dimensional characterization of high aspect ratio MEMS using only two-dimensional data.

CHAPTER II

LITERATURE REVIEW AND BACKGROUND

High Aspect Ratio MEMS Fabrication

The motivation for this research stems from the relatively new research field of MEMS (micro-electromechanical systems). Specifically, parts with high aspect ratios, such as those manufactured from the LIGA micro-fabrication process, are of interest. One of the main factors preventing these parts from commercialization is the lack of tools for analyzing the dimensional quality of MEMS, especially those with steep sidewalls. A brief introduction to micro-fabrication follows and then specifics about the LIGA process are discussed. Then, the current state of metrology for these parts is presented to highlight what is lacking in this field of research. Based on the literature review, initial conclusions are drawn about the need for full three-dimensional data of high aspect ratio parts.

Micro-fabrication

Photolithography is by far the most prevalent lithography technique used in micro-fabrication. Due to its relatively inexpensive nature and flood exposure capability, photolithography dominates as the fabrication method for MEMS. Photolithography can

typically produce accurate parts with heights of up to a few μm. After heights reach greater than 10 μm, light scattering and depth of focus (DOF) issues limit the edge definition of parts that can be fabricated by photolithography (Hartley et al. 1998). This limitation severely restricts the range of actuators and sensors that can be accurately produced when high aspect ratios are needed. X-ray lithography is a technique that overcomes many of the limitations of traditional photolithography.

LIGA Fabrication

The LIGA process is a fabrication method used to manufacture high aspect ratio MEMS. This process was initially developed in the late 1970s at Forschungszentrum Karlsruhe in order to manufacture small slotted nozzles for uranium isotope separation (Becker et al. 1982 and Bley et al. 1984). The name is a German acronym for lithography (lithographie), electrodeposition (galvanoformung), and molding (abformtechnik). LIGA allows for the fabrication of relatively tall structures (up to several mm) with parallel sidewalls. The primary characteristic of the LIGA process is the use of X-rays as the exposure source. The process is not limited by many of the factors that affect traditional UV micromachining techniques such as depth of focus and high photon scattering. Another advantage of X-ray lithography over photolithography includes higher lateral resolution (0.2 μm vs. 2 μm) (Brodie et al. 1982). The primary motivation for this research stems from this fabrication method.

The LIGA process typically starts with the creation of a mask. There is no current standard on LIGA mask fabrication because of the design parameters that vary from part batch to part batch (e.g., feature size, part height, tolerances, spectrum of synchrotron radiation, etc.). In practice, there are generally two methods used for mask fabrication

(Hruby 2001). Membrane type masks are usually needed for low beam-energy setups or fine feature sizes. Membrane masks use a thin, X-ray transparent base material to avoid any scattering of the X-rays. Substrate masks are thicker and more robust than membrane masks, but are limited in the transferable feature size to the X-ray resist. Increases in exposure time are also induced when using substrate masks (Griffiths et al. 2000).

The mask is typically plated with a high atomic number element such as gold which effectively blocks X-rays. Once the mask is completed, the substrate is prepared using a silicon wafer as the base. Thin layers of Ti and Cu are sputtered onto the substrate for adhesive and conductive purposes. Next, an X-ray resist is spin-cast onto the wafer. This resist is typically polymethylmethacrylate (PMMA), a well-known X-ray photoresist. Other resists are currently under research and a resist has been developed specifically for X-ray lithography which claims 15x the sensitivity of PMMA (Schenk et al. 1997). The resist height can range from hundreds of μm up to several mm (Madou 1997). The substrate is then exposed to X-rays to pattern the resist. Synchrotron radiation sources are the primary source for X-ray generation. Synchrotron sources produce the most usable amount of collimated X-rays which allow for shorter exposure times and higher throughput. Figure 2 illustrates the typical setup of a membrane mask and substrate in a synchrotron source.

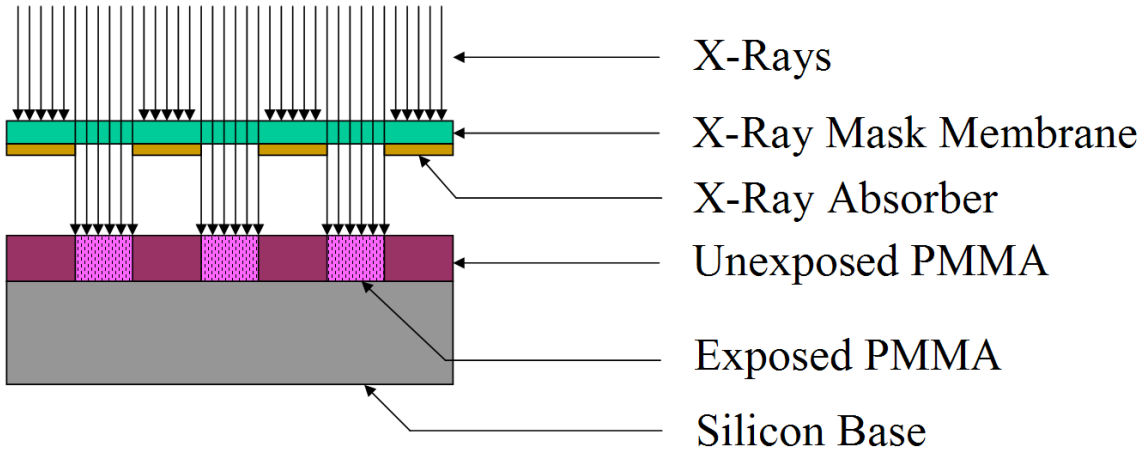

Figure 2: X-ray exposure process

Once the PMMA is exposed, it is developed with an organic solvent leaving the patterned unexposed PMMA. This pattern can then be electroplated with various metals. Nickel is typically used because of its controllability in the electroplating process. Models do not currently exist, however, to predict the electroplating process and this step remains as a major source of geometry variation (Hruby 2001). Most research relies on past experience to accurately electroplate the substrate. Once the pattern has been electroplated, the surface is planarized using lapping techniques, and then the unexposed PMMA is dissolved using another chemical solution. The free standing metal structures remain and can either be released from the substrate as individual parts or used as a mold for precision injection molding. Depending on various process parameters, the geometries of the free standing structures can vary. There are three general cases common to most MEMS, as depicted in Figure 3. These particular cases are primarily a factor of exposure time.

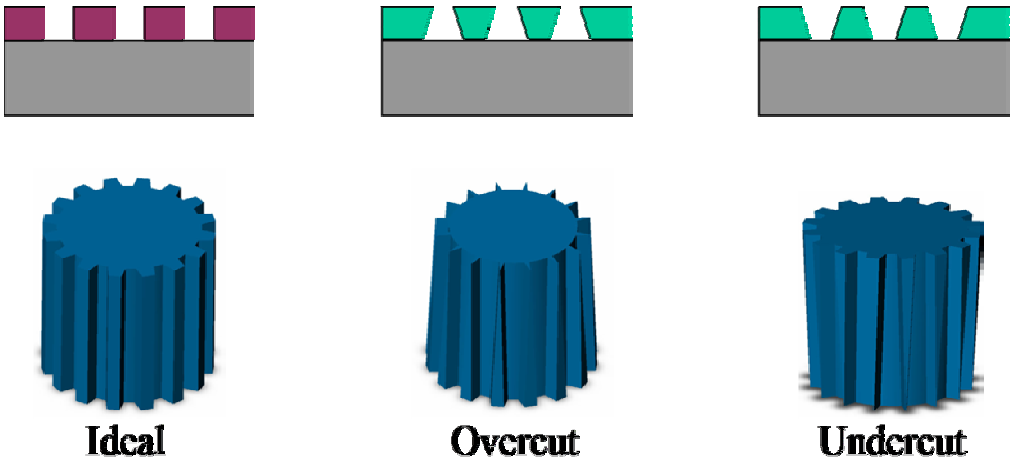

Figure 3: Final part geometry based on various process parameters

The LIGA fabrication process requires the use of X-ray radiation as described above. Due to the inherently expensive equipment used to generate X-rays, there are few facilities that can fabricate these high aspect ratio devices. With limited facilities, it follows that research in this area is also limited. Currently, the fabrication process itself is continually being modified to produce parts with the desired material and geometric properties. Metrology of these high aspect ratio parts is in its infancy, as the three-dimensional nature of these parts proves significantly difficult to analyze with traditional semiconductor metrology hardware.

MEMS Metrology

Methods for MEMS metrology are primarily derived from techniques of the semiconductor industry. The characterizing term traditionally used for the semiconductor industry is the critical dimension (Larrabee and Postek, 1994). A critical dimension (CD) is typically represented by a line width or pitch width measurement. These

measurements are inherently one-dimensional and provide minimal feedback with respect to the overall part geometry. However, semiconductor metrology techniques are frequently the only feasible method for obtaining geometric information from a MEMS part, and thus the most prevalent data acquisition techniques will be discussed here.

Optical microscopes are used to inspect relatively large MEMS such as those fabricated from the LIGA process, which has the capability of producing parts that are a few mm tall. The underlying operating principles for optical microscopes include spatial resolution that is determined by the Rayleigh criterion and detected edge sharpness that is determined by a combination of hardware (e.g. lens type, CCD camera) and lighting conditions (e.g. coaxial lighting, ring lighting).

Optical microscopes have the advantages of being fast and non-destructive. Rarely do test parts have to be modified (e.g. coated with a conductive material) from their original form. Optical microscopes tend to be repeatable for features as small as 0.25 μm. The ultimate limiting factor for resolution of optical metrology hardware is diffraction and the ability of the microscope to produce images with sharp edges in order to accurately detect edges. Other significant errors of optical techniques typically stem from interference, resonance, shadowing, and lens distortions (Sheats 1998). Ceremuga (2003) characterizes an optical microscope which is heavily used for MEMS inspection, and his results show good agreement with the stated specifications of the machine. Other topics analyzed in the work include choosing the best location on intensity curves of optical microscopes to determine the actual location of an edge.

An important limitation of optical microscopes for MEMS inspection is inability to acquire true three-dimensional data. Some optical microscopes are integrated with

software that uses image processing techniques to determine the Z-height at which the scan is taking place. The current state-of-the-art software uses a projected Ronchi grid to determine the height at which the microscope is focused in one region of the image (View Engineering 2003). If the region selected has multiple focus points (i.e., the region selected in not all on one plane), the algorithm assigns the average value for the Z-height. Further edge detection algorithms are run to extract X and Y data from the microscope image. This technique, in theory, produces three-dimensional data from an image; however, the algorithms used after finding the Z-height in one location of the image assume that all of the data are on the same plane. Thus, the data acquired from vision systems such as these can be characterized as acquiring 2.5-D data sets.

One of the primary tools used for analysis of MEMS devices is the scanning electron microscope (SEM). SEMs are capable of producing high resolution images on the angstrom scale. Electrons were first shown to exhibit wave properties proportional to an applied electrostatic potential difference by Davisson and Germer in 1927 (Sheats 1998) and the first SEM used to examine the surface of a specimen was at RCA in the United States (Zworykin et al. 1942). Postek (1994) provides a thorough analysis of the capabilities of SEMs. To summarize, the accuracy of the images captured is highly dependent on machine capability and the specific part being examined. Beam-sample interactions (i.e. charging) are shown to greatly affect the results of any measurement taken with the device. Additionally, despite the high resolutions of the SEM, the output is typically generated from the electron detector and displayed on a cathode ray tube rastered in synchronization with the electron beam. The final result is a two-dimensional image on a screen. Since no coordinate data are directly outputted from the SEM,

performing any analysis other than line width measurements directly with the SEM software becomes difficult. Thus, SEMs are ideal for visualizing MEMS parts, but are inadequate tools for quantitative analysis of MEMS devices.

Other metrology hardware exists which can capture full three-dimensional data. Scanning white light interferometry is a technique used frequently to capture surface roughness of small parts (Wyant 2002). White light interferometry is a relatively quick, non-destructive, non-contact method to inspect surface roughness. It is a multi-wavelength technique that relies on a spectrally broadband source to eliminate fringing order errors common to monochromatic interferometers (Roth, Felkel, and Groot 2003). White light interferometers have sub-nanometer resolution in the scanning direction (i.e. z-direction) and can be used on a multitude parts with different surface finishes (Groot and Deck 1994). These tools have resolutions down to the sub-nanometer, but are limited to the degree of slope of the surface. The largest slope that can be identified is typically around 30 degrees with a 100x objective (Zygo 2002). As the objective power decreases, the identifiable slopes also decrease. Since LIGA parts typically have near vertical sidewalls, hardly any three-dimensional data from the sidewalls can be accurately captured. Despite these limitations, white light interferometry is heavily used in the MEMS industry to determine surface roughness characteristics of parts (Mahoney et al. 2003).

Scanning probe microscopes (SPMs) offer another alternative to non-contact techniques. SPMs are characterized by high resolution (sub-angstrom to angstrom). The two most widely used SPMs are the scanning tunneling microscope (STM) and the atomic force microscope (AFM). The older of the two technologies is the STM which

was developed in 1982 (Binnig and Rorher 2000). This technique uses a metallic probe that is brought into close proximity of a conductive surface so that a small current flows between the probe and surface. The current is held constant by a feedback control scheme, and the probe then tracks the height of the surface (Sheats 1998). Sub-angstrom resolution is attainable in the normal direction of the surface, and angstrom-scale resolution is attainable in the lateral directions of the surface.

Atomic force microscopy (AFM) is the newer SPM technology that retains the resolution of the STM, but is not limited to conductive surfaces (Binnig, Quate, and Gerber 1986). The measurements of an AFM are performed with a sharp probe that collects a series of line scans across the surface of a part. The topography of the part is measured by bringing the probe close to the specimen and measuring the repulsive and attractive forces on the probe tip. In the Z-direction, AFMs have high sensitivity, typically having a resolution of 0.05 nm. Resolution in the X and Y-directions are also high, ranging from 2-10 nm (Veeco 2003).

There are certain limitations to SPMs, particularly in measuring high aspect ratio parts. STMs, as previously mentioned, are limited to parts with conductive surfaces. All SPMs are limited, in the same sense as white light interferometers, to the maximum angle between two surfaces of a part. When features with perpendicular sidewalls are scanned, the data typically exhibit a slope or curtain that is actually not present (Griffith, Marchman, and Hopkins 1995). The height of parts is also limited to the probe length which is typically limited to 10 μm. This limitation severely prohibits the inspection of high aspect ratio parts with dimensions on the order of mm. Though these tools have

extremely high resolution, it is unfeasible to collect scans that cover all of the surfaces of a part, given the limited scan range of the tools.

Current research is trying to address some of the limitations of current metrology hardware for MEMS. Micro-interferometry is one area of current interest. Using a coherent light source, micro-lenses, and micro-diffraction gratings, these devices have demonstrated measurement capabilities on the scale of nm (Kim et al. 2002). The proposed technology has the advantage of being able to make both static and dynamic measurements (Kim et al. 2003). Hall and Degertekin (2002) show initial results for the displacement measurements of acoustic transducers. These tools are currently limited to changes in step height proportional to the wavelength of light used in the laser source, which limits the range of measurement for the device.

Small-scale coordinate measuring machines (CMMs) are also being investigated for their possible use in inspecting MEMS parts (Peggs 1999). These devices have a 50 x 50 x 50 mm measuring volume and target sub-micrometer uncertainties (Peggs et al. 2003). The main issues yet to be addressed are the size, quality and calibration of the probe tip used for inspection. The smallest size probes to date are on the order of 0.1 mm (Schellekens and Haitjema 2001). In addition, the design of a sensing system to detect the small displacement forces of the probe proves to be a challenge.

X-ray tomography is another area of current research. This technique assembles 2-D images taken from X-ray scans of the object into 3-D views. X-ray tomography has the capability of capturing dense sets of data, but does not currently possess the necessary resolution to effectively analyze MEMS devices. These systems currently have resolutions on the order of μm (Aracor 2002), which is on the same scale of the parts

being measured and therefore not detailed enough for effective measurements. Lawrence Livermore has also started a program to design an X-ray tomography system to measure meso-scale (i.e. features ranging from μm to mm) parts (Heller 2003).

The use of these tools to characterize high aspect ratio MEMS has met with limited success. Prasad et al. (2000) show interesting results using an SEM to visualize LIGA sidewall geometries. For a series of Cu parts fabricated using the LIGA process, the lapping procedure is shown to have the greatest effect on the final sidewall geometry of the part. During the lapping process, the Cu is shown to plastically deform at the edges causing a "foot" on the sidewall. This process detrimentally affects the linear sidewall geometry of a part. For Ni parts fabricated with the LIGA process, no visual deformation due to lapping was noticeable. However, scales were noticeable in the sidewall images of Ni gears resulting in non-smooth sidewalls. Other regressions noticed in various LIGA parts include incomplete removal of polymer film and sidewall roughness induced from mold surfaces. The exact mechanisms, whether from processing steps or the microstructure of the material, for the cause of many of these regressions are still unresolved.

In summary, the ability to fully quantify the dimensional characteristics of high aspect ratio MEMS is currently non-existent. Hibbard and Bono (2003) have started a survey of current tools available for MEMS metrology and initial findings show there is a significant gap in capabilities for MEMS inspection. Tools exist that have exceedingly high resolution, but are limited in range (i.e. only a few square μm can be measured). Other tools exist which have sufficient range, but are limited to drastic slope changes on the surface of the part, a characteristic typical of high aspect ratio MEMS. The current

solution to quantifying meso-scale MEMS parts is to use a combination of tools, particularly vision-based systems and white light interferometers. The combination of these techniques, at best, provides form error information on the edges of the part and surface roughness data on the sidewalls of the part. However, no information is provided regarding the overall geometry of the part. One conclusion that is made here is that sidewall geometry of LIGA parts can vary based on numerous known and unknown issues. Until the LIGA fabrication process is optimized to consistently produce straight sidewalls with minimal post-processing induced errors, inferring the sidewall geometry from scans of only the top and bottom of the part is not reliable.

Computer-Aided Inspection

The overall goal of the hardware aspect of coordinate metrology is to produce a digitized representation of the geometry of a part. Once the data acquisition step has been completed, the data must be processed in order to provide useful feedback about the geometry of the part. Typically, the data is imported into a computer-aided inspection (CAI) program. The main objective of these programs is to take the unprocessed coordinate data and compare them to the nominal CAD geometry.

Current stand-alone metrology software packages are similar to one another in design. There are two general types of metrology software packages, based on the input type of the design and data. Metrology packages are either based on 2-D CAD models (e.g., AutoCAD) and two-dimensional data (e.g., X-Y data) or based on 3-D CAD

models (e.g., Solidworks) and three-dimensional data (e.g., X-Y-Z data). Most of these packages assume the following work flow for computer-aided inspection (Claudet 2001).

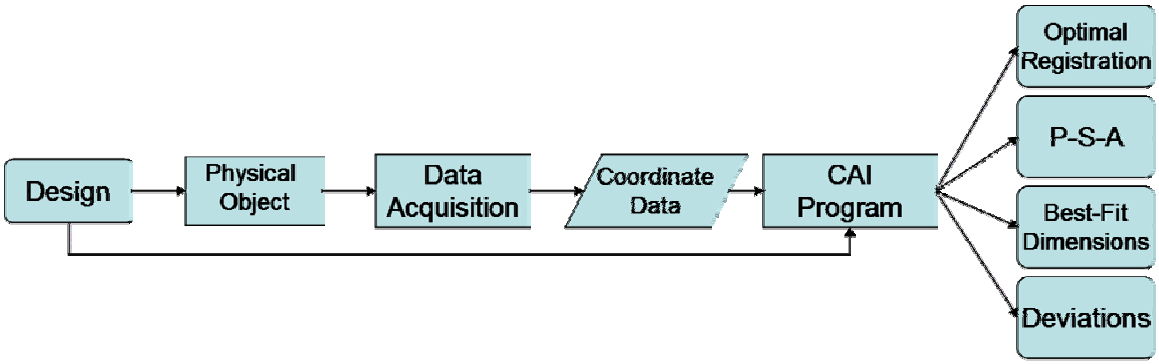

Figure 4: Work flow in computer aided inspection

When the CAD design and the physical data reach the CAI package, various processes occur within the software. First and foremost, the CAD model and data must be aligned to the same coordinate frame. This situation is typically posed as a least-squares minimization problem which can be solved by various methods. The objective function for the least-squares problem is typically stated in the following form:

$$f = \min \sum_{i=0}^{n} \|e_i\|^2 \qquad (1)$$

In the sense of data localization, the error term, e_i, represents the normal distance from the point in question to the surface of the model. The normal distance, however, cannot just be a static calculation in the standard sense. The e_i terms must come from an objective function that allows manipulations of the point in 2-D or 3-D space. A standard

way to manipulate data points is a homogenous transformation (Mortenson 1997) as shown in Equation (2). The homogenous transformation shown in Equation (2) represents translations along the three principal axes and three rotations about the principal axes, respectively.

$$\vec{P}^* = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \left[\vec{P}\right] \quad (2)$$

Though least-squares minimization is not theoretically always the appropriate fit method, this method has, in practice, prevailed as the preferred choice. Choi (1997) details cases when least-squares fit should be avoided and other methods such as a zone fit should be used.

The currently accepted state-of-the-art algorithms used in commercial metrology packages are typically based on the Iterative Closest Point (ICP) method developed by Besl and McKay (1992). The ICP method uses the least-squares criterion for termination; however, the method does not use homogenous transformations for the objective function parameters as in Equation (2). Instead, Besl and McKay utilize principles of quaternions to find the best-fit parameters of a data set to a surface. Other variants of the ICP method have been developed in order to increase efficiency (Rusinkiewicz and Levoy 2001). Work by Tucker (2000), however, shows that even though the ICP method is robust, it is not a true minimizer of the least-squares problem.

Other methods for registration have been developed to register data to CAD surfaces. Sahoo and Menq (1991) present a method for localizing complex sculptured surfaces. They deal with planar, quadric, and higher order polynomial surfaces. The sum

of the squared distances from the point cloud to the model are minimized with respect to six transformation parameters as shown in Equation (2). The minimization procedure used for Sahoo and Menq's research is a combination of a Steepest Descent Method and a Newton Raphson Method. The Steepest Descent Method converts the minimization into a one-dimensional problem and when convergence is close, the Newton Raphson Method is implemented for final convergence. Claudet (2001) takes a similar approach to the minimization problem, but derives deviation functions and analytical gradients that are not approximations such as those used by Sahoo and Menq. Claudet's algorithms lead to a more robust and accurate localization. Both Tucker and Claudet provide case studies which actually show that Newton methods converge faster and more accurately than the ICP method when given a good initial guess of the transformation parameters.

Before the localization routine can be implemented, though, each data point must be assigned to a corresponding geometric entity of the CAD model to which it supposedly belongs. This step is, appropriately, called point-to-entity assignment. A CAD model can have thousands of faces and/or edges, and before a data point can be registered, it must be assigned to the appropriate entity. This task typically involves the most computationally intensive algorithms since every data point is usually assigned to a geometric entity for each iteration of the minimization. Point-to-entity assignment can be performed only once at the beginning of minimization, but as the data is transformed to the CAD model, there is no guarantee that each data point belongs to its originally assigned surface. Point-to-entity assignment is also one of the main reasons for the two different types of metrology software packages mentioned above. A simplified topology

graph of a typical CAD model is shown in Figure 5. The basic definitions of the entities that compose a CAD model are as follows (Spatial 2002):

1. Body - The highest level of model object.
2. Shell - A set of connected faces which can bound the outside of a solid.
3. Face - A connected portion of a surface bounded by a set of edges.
4. Edge - A curve bounded by vertices.
5. Vertex - A point typically denoted by a triplet of numbers (i.e., X, Y, Z)

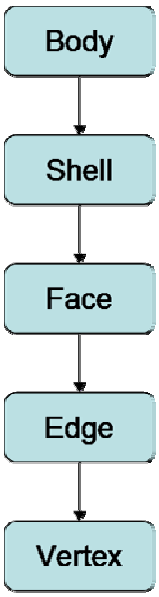

Figure 5: Simplified topology of a CAD model

Metrology packages based on a 2-D framework only deal with edge geometry and two-dimensional data. During the point-to-entity step of registration, a 2-D framework only analyzes the edges of a CAD model and the associated data. On the other hand, packages based on a 3-D framework deal only with face geometry, assuming the data are representative of the faces of a CAD model. In the point-to-entity step, only faces are considered in the analysis. Some work has been done on different procedures for point-

to-entity assignment such as exhaustive and plane sweep assignment for surfaces (Claudet 2001). In the exhaustive search, every point is compared with every surface which is simple and robust. A plane sweep approach generates a list of events to more efficiently determine which points belong to which surfaces. The idea here is to sweep an imaginary plane through the entire scene, which includes both the data and the CAD model, and events are created as the plane intersects a point or a surface. A point event immediately followed by a surface event typically indicates that the data point is closest to that surface. This method is typically more efficient than an exhaustive search.

An issue of relatively recent attention has been the accuracy of the algorithms used in data analysis software. No current standards exist that regulate the algorithms used in data analysis. Walker (1988) issued an advisory in which he stated that certain algorithms are capable of introducing an error of up to 50%. Estler (1989) analyzed a measurement device for NASA and concluded that the data analysis software was the single largest source of error in the entire system. Hopp (1993) discusses the validity of results generated by the fitting objectives in minimization problems. He points out that there are no standards that computational metrology algorithms must conform to, and thus, there are significant variations in the reported deviation results between algorithms for the same data sets. The National Institute of Standards and Technology has developed a testing service that quantitatively measures the performance of fitting software. They use data sets of known error and compare the results to that of the fitting software. This service acts as a calibration measure for testing fitting algorithms. Hopp (1995) believes this NIST service will serve as one of the much-needed requirements for

fitting algorithms to adhere to in order to reliably report deviations. Shakarji (1998) details the methods NIST uses to evaluate software.

The issue with current metrology packages when applied to the MEMS field is the decision to use a 2-D or 3-D platform. This problem limits the user to the types of models and data that can be analyzed, when in fact both types of analyses are needed based on the type of data acquisition method and the type of CAD model. This is one limitation that this research intends to address.

After the data have been localized to the CAD model, various analysis routines can be implemented to find tolerances and other fit parameters. Figure 6 shows an initial scene with the unregistered data, then the registered data with a plot of the deviations to the surface. Other analyses include determining the best-fit parameters of a data set. For example, if a set of data represent a cylinder, a minimization can be performed on the data to determine the best fit cylinder to that data, without the need for a CAD model.

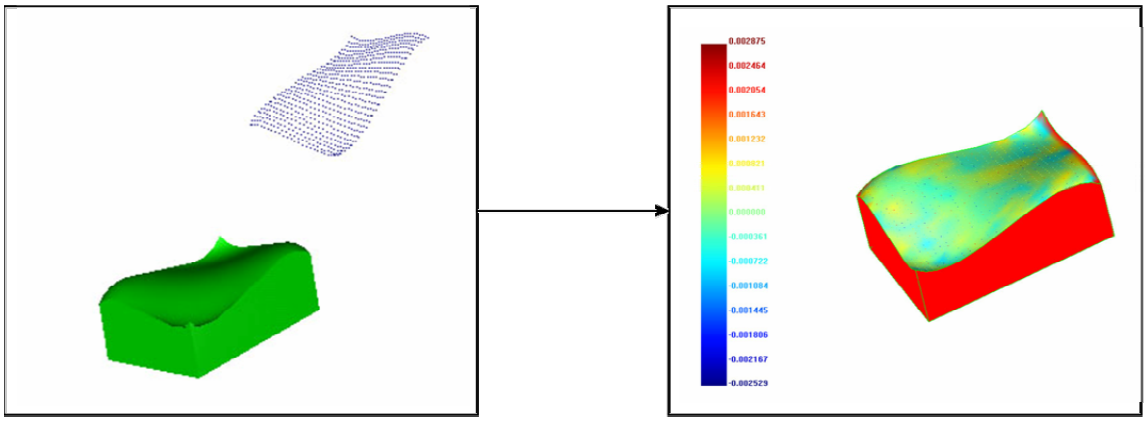

Figure 6: CAD model before and after localization

In summary, there are two disparate types of CAI software packages currently available, those for analyzing 2-D data and those for analyzing 3-D data. Because multiple data acquisition techniques that output either 2-D or 3-D data are employed to analyze MEMS devices, this limitation is inconvenient. This research intends to address this problem by developing a flexible framework to incorporate any type of needed analysis, independent of the type of data and CAD models. Also, there is clearly a need to test and analyze the accuracy of algorithms implemented in any type of data analysis software as shown by the results in the current body of literature. The algorithms developed in this research will conform to the suggestions of the NIST algorithm testing service, and each algorithm will be tested to quantify the validity of the results.

CHAPTER III

EQUIPMENT AND INSTRUMENTATION

Overview

Throughout the course of this research, various tools were used to design and validate the methods developed in this project. Descriptions of the primary tools used in this research follow. First, the software tools used to implement the algorithms and methodologies are discussed. Emphasis is placed on the primary reasons these tools were chosen and why they are appropriate to the problem at hand. Then, fabrication tools excluding the LIGA process, which was described in the previous chapter, are discussed that were used to create validation case studies for the software developed. Finally, the data acquisition tools used to acquire the experimental data are covered, with emphasis placed on the limitations and precision of the machines in order to provide a sense of fidelity for the experimental data gathered.

Software

Microsoft Visual C++

Microsoft's Visual C++/MFC is a development environment that encapsulates the C++ language. The C++ language was chosen because of its extreme flexibility and computational speed. C++ is an object-oriented programming language which allows for the development of clearly defined objects which combine to create large, yet manageable, programs. C++ is well suited for the research topic at hand because of its ability to quickly manipulate large data sets and perform complex calculations efficiently. One last reason C++ was chosen was for the wide variety of toolkits available for scientific and engineering computing.

Besides the code compiler, Microsoft Visual C++ consists of a set of C++ base classes that enable the efficient design of user interfaces native to the Windows format. These classes make up a library called the Microsoft Foundation Classes (MFC). These base classes allow for the design of software that has the look and feel of standard Windows programs in order to reduce the learning curve for the end user. The Visual C++ development environment was used to implement all the algorithms and user interfaces for this dissertation.

Spatial ACIS Modeling Kernel

ACIS is an object-oriented three-dimensional modeling engine from Spatial Technologies. The ACIS toolkit consists of a set of C++ classes that serve as a unified environment for geometric modeling. ACIS integrates wireframe, surface, and solid modeling into a unified data structure. ACIS is a boundary–representation (B-rep)

modeler, which means that it defines a boundary between solid material and empty space. ACIS can represent both two-dimensional and three-dimensional models and data which is a critical requirement of this research. Using ACIS, one can query a model for its geometry and retrieve, for instance, the mathematical equation of a surface. This mathematical equation can then be used to compute the distance from the surface to a point in space as typically done in a computer-aided inspection program. ACIS makes any of these types of analyses easier than if one had to manually program geometry and topology information into a software package. Also, the file type used by ACIS (.sat) is a widely used format in industry, thus making it easy to incorporate the analysis of real world CAD models. Many of the algorithms developed for this work utilize ACIS classes in order to represent geometric entities.

Techsoft HOOPS Visualization Kernel

The HOOPS visualization kernel is set of toolkits useful for visualizing geometric models and point cloud data. The HOOPS kernel provides a high level API for developers to program highly optimized visualization programs in shorter development times than if using a lower level toolkit such as OpenGL. The HOOPS kernel provides a bridge that facets ACIS models and displays them to the user. HOOPS also contains a model/view/operator (MVO) library which allows for user interaction of models and application-specific data. For this research, HOOPS was used to display all geometric entities as well as for user interaction with the CAI scene.

Hardware

Giddings & Lewis Fadal VMC 15

Since the tools for MEMS metrology are lacking, several macro-scale case studies were devised in order to validate the algorithms developed in this research. The Fadal VMC 15 is a vertical machining center with the following relevant specifications (Giddings 2004):

Table 1: Fadal VMC 15 Specifications

Table Size	74.93 x 40.64 cm (29.5 x 16 in)
Accuracy, Axis Positioning	+/- 0.0051 mm (0.0002 in)
Accuracy, Axis Repeatability	+/- 0.0025mm (0.0001 in)
Spindle Speed	10 – 7,500 RPM
Spindle Horsepower	15 HP
Machine Controller	Fadal 32MP

The Fadal machining center is used to manufacture test parts for inspection on a coordinate measuring machine. A photograph of Fadal VMC 15 is shown in Figure 7. The material used for the parts was aluminum because of its ease of machinability. Surfware's SurfCAM software was used to generate the appropriate CNC code for the machine. A 4" fly-cutter, 1" end mill, and 3/8" ball-nose mill were all used to fabricate the part. The final cutting pass for the test parts used a 0.0076 mm step height in the Z-direction for an acceptable surface finish.

Figure 7: Giddings and Lewis Fadal VMC 15

3D Systems Viper si2 SLA

The 3D Systems Vyper si2 SLA system is a stereolithography apparatus (SLA) which builds 3D parts layer-by-layer using a photo-cured resin. The Viper is used as another manufacturing method for the test parts. The relevant specifications for this machine are shown in Table 2 (3D Systems 2003).

Table 2: Viper si2 Specifications

Laser	354.7 nm solid state Nd:YVO_4
Beam diameter	Standard mode: 0.250 mm (0.010 in) High res mode: 0.075 mm (0.0030 in)
Vertical resolution	0.0025 mm (0.0001 in)
Position repeatability	0.0076 mm (0.0003 in)
Typical velocity during part build	5 mm/sec (0.2 in /sec)
Max. build envelope	Standard mode: 250 x 250 x 250 mm (10 x 10 x 10 in) High res mode: 125 x 125 x 250 mm (5 x 5 x 10 in)

The Viper SLA system is used to make the same test parts as with the Fadal VMC. A step height of 0.0076 mm was used for the build layers. A photograph of SLA used to manufacture the test cases is shown in Figure 8.

Figure 8: 3D Systems Viper si2 SLA

Brown & Sharpe MicroVal PFx CMM

A coordinate measuring machine is used to acquire data from the parts. As previously mentioned, a CMM is a metrology device which uses a touch-trigger probe to acquire data. The Brown & Sharpe MicroVal PFx CMM (Figure 9) is a vertical bridge type machine which operates in manual, fine positioning, or direct computer control (DCC) mode. All data for this research is collected using DCC for the highest possible accuracy. The relevant specifications for this machine are as follows (Brown & Sharpe 1991):

Table 3: MicroVal PFx Specifications

Measuring Envelope	457 x 508 x 406 mm (18 x 20 x 16 in)
Repeatability B89	0.003 mm (0.00015 in)
Volumetric Accuracy B89	0.010 mm (0.0004 in)
Linear Accuracy B89	0.005 mm (0.0002 in)
Resolution	0.0001 mm (0.00004 in)

Figure 9: Brown and Sharpe MicroVal PFx

View Voyager V6x12 Vision-based Metrology System

The View Voyager is a non-contact, dimensional metrology system designed for use in high volume production environments that require high throughput. The system uses advanced edge detection techniques as well as patented auto-focus algorithms to accurately acquire 2.5-D part data. Table 4 gives the specifications of the machine (View 2003).

Table 4: Voyager V6x12 Specifications

Measuring Range	305 x 152 x 152 mm (12 x 6 x 6 in.)
Resolution	Standard: 0.1 μm (0.000004 in.)
Measuring Accuracy	$U_{2\ (XY\ plane)} = (3.5 + L/300)$ μm $U_{1\ (Z\text{-}axis)} = (3.0 + L/50)$ μm Where L = measuring length in mm.
Stage Drive System	DC servo motor control
Illumination	Standard: Programmable backlight and coaxial surface light Optional: Multi-color, 4-quadrant, LED Ring Light
Optical Sensor	Digitally processed instrument quality monochrome CCD camera with 756 x 581 pixel array.
Image Processing	256, 8-bit, grayscale gradient processing Programmable filtering and outlier point removal. Continuous point edge finding tools, area centroid, histogram, blob (defect), and autofocus tools.

The Voyager (Figure 10) is used in this research to acquire all edge data from the LIGA case study part. All data gathered from the Voyager comes directly from the edge detection algorithms in X, Y, Z text format.

Figure 10: View Voyager V6x12

CHAPTER IV

FORMULATION AND METHODS

Overview

This chapter details all of the mathematics and algorithms developed in this research. The first topic covered is the geometrical entities that form the foundation for the analysis procedures developed. Since most high aspect ratio fabrication processes only addresses analytic shapes for the design of the parts, these geometries are addressed more rigorously mathematically for computational efficiency. In order to ensure a certain level of robustness, freeform geometries are discussed and methods are implemented for their analysis, but most of the intensive calculations are left to the geometry kernel as opposed to the derivation of the mathematical formulations directly. No claim towards their efficiency can be made because of the proprietary nature of the kernel.

The second section formulates the methods to manipulate point clouds used in this research. A general approach is taken similar to that of Claudet (2001) which enables the black-box design of transformations that can simply be concatenated together to create procedural transformations for the data.

The third section addresses the mathematics derived to perform point-to-entity calculations. In any CAI program, one of the main tasks to be completed is the calculation of deviations from a point cloud to a geometric entity. These deviations

typically serve as the input to the registration procedures as well as the standard output for post-processing of the data.

Since most CAD models are comprised of more than one entity (e.g. multiple edges, multiple surfaces, or both), point-to-entity assignment is typically necessary before deviations can be calculated. The methods for point-to-entity assignment chosen for this research are derived and discussed in the fourth section.

In the next section, the algorithms used for cloud-to-CAD registration are discussed. Theory in optimization and minimization is typically considered mature, so an established, robust method for minimization is chosen and developed. Minimization is not always the desired method for point registration, and a fiducial based registration process is developed to address those cases. The fiducial based algorithms developed in this research are critical to the analysis of multiple point clouds to one CAD model and are discussed in this section. The fiducial based analysis serves as a key component to the analysis of high aspect ratio MEMS.

The sixth section of this chapter discusses routines for geometric parameter fitting, i.e. fitting data without a CAD model. These analyses are useful for cases when a CAD model is not available and for fiducial based registrations. This problem turns out to be similar to the minimization problem with a CAD model, but with minor changes. The framework developed in this research allows the easy extension of the point to entity derivations in order to perform geometric parameter fitting.

The final section details the output of the registration process. Typically, deviations from the point cloud to the ideal geometry are exported for further statistical

analysis, or some type of visualization is implemented to provide visual feedback of the computed errors in the part.

Geometry Representation

The main foundation of any type of part analysis is the underlying geometry which represents the part. The majority of industry relies on information stored in two-dimensional CAD models to represent ideal physical parts. The advantages of three-dimensional modeling are being realized, and many industries are moving towards this design paradigm.

As previously mentioned, commercial CAI programs come in two forms: those based on a two-dimensional approach and those based on a three-dimensional approach. This research investigates combining these two approaches to allow for complete flexibility in the inspection process. The main advantage to combining these two design philosophies is to enable complete independence of the data acquisition hardware. Since both two-dimensional and three-dimensional data acquisition tools are used for MEMS metrology, a single platform that allows for both types of analyses is significant. The primary difficulty in developing a combined analysis platform is the proper handling of both two-dimensional and three-dimensional geometries during registration of point data.

Choosing the ACIS modeling kernel and C++ as the development tools for this research allows for the efficient and logical organization of geometric data. A powerful feature of C++ is class inheritance which allows for a general "parent" class to be developed with some base functionality which all "child" classes inherit. Then, if these

base functions are declared *virtual*, the child class can override the base functions to make it specific to that particular child class. This feature allows for other functions to operate on the general instances of a class (i.e. parent class), but having the specific child instance of a function be called if necessary. For example, suppose a base class called Geometry is defined that has a *virtual* function called ReturnName(). For the Geometry class, assume ReturnName() returns the string "General Geometry." Now, suppose a class called Line is derived from Geometry. This class inherits the function ReturnName() and, since it is declared *virtual*, the Line class can redefine this function to return the string "Linear Geometry." Hypothetically, let there be a function that takes a Geometry class and prints out the name of that instance. Since the Line class is derived from Geometry, an instance of Line can be passed to this function and the correct ReturnName() function is called. This feature is powerful from an implementation standpoint because functions can be developed that take any general class, but permit redefined functions of a derived class to be called. In this research, class inheritance and *virtual* functions serve as key enabling features for the representation of geometry.

As previously mentioned, the LIGA process is a 2.5-D manufacturing process; thus, all of the CAD models used are 2-D. These 2-D shapes are currently limited to straight lines and arcs, due to the limitations in fabrication equipment. This aspect of the fabrication process is not likely to change in the near future, simply because of the fact that any geometry can typically be simplified and represented with these basic shapes given the desired tolerances. Most of the metrology tools for MEMS devices are also 2-D in nature, such as SEMs, traditional microscopes, and vision-based systems. The trend here, though, is different. As these tools become more advanced, acquiring 3-D data

becomes desirable in order to provide more dimensional information about the part. In order to facilitate the advancement of 3-D metrology tools, this research investigates the handling of both 2-D and 3-D geometry for part analysis.

For this research, the topologies of edges and faces are of primary importance (Figure 5). The geometries associated with the topology of edges are used to compute deviations to planar data, and the geometries associated with faces are used to compute deviations to surface data. The geometries implemented in this research include the following: straight line, circular curve, general parametric curve, plane, sphere, cylinder, cone, torus, and general parametric surface.

The other geometrical entity heavily used in this research is a point cloud. A point cloud is typically representative of acquired data from metrology hardware. Ideally, these points correspond to points lying on an edge or face of a CAD model. However, due to inaccuracies in the manufactured part as well as errors induced by data acquisition, some errors between the points and their respective edges or faces will occur. Therefore, a method is needed to compare these points to the nominal geometry. This situation is typically posed as a least-squares problem with the deviation of a point to the CAD model serving as the objective of minimization. Before the least-squares problem can be solved, a method to transform the points or the CAD model to their best-fit location must be developed. This is the topic of the next section.

Transformations

There are two options when registering a CAD model and point cloud data: moving the CAD model to align with the point cloud, or moving the point cloud to align with the CAD model. Since the CAD model must be transformed with the proprietary functions of the modeling kernel, it is easier and typically more efficient to transform the raw point cloud data.

In traditional geometric modeling, rigid-body transforms are used to move entities in 2-D or 3-D space. Rigid-body transformations are the most restrictive of affine transformations, where all metric properties are invariant (e.g. distance, angle, etc.) Dilation (i.e. scale) transformations are also affine transformations which preserve angles, but not necessarily the other metric properties (Mortenson 1997). These are the types of transformations on which this research focuses. Because of the extensibility of the framework developed in this work, other transformations are easily introduced if so desired.

As mentioned in Chapter II, the standard way of representing transformations is by the use of a 4x4 homogenous transformation matrix. Matrix operations are a clear and concise means to transform any given object, but are inefficient programmatically. This work builds on procedural transforms introduced by Claudet (2001). These transforms fit in well with the programming paradigms of an object-oriented language such as C++ and result in an efficient and elegant way to manipulate point clouds.

Transformations are inherently procedural when applied to an entity. An object that is first rotated by θ and then translated by x is not in the same position as one translated by x then rotated by θ. A procedural transform simply takes a point P and

maps it to a different position, P'. One property of the procedural transform is the ability to concatenate a list of transforms. This property allows for the design of simple, object-oriented transforms (e.g. translation in x, rotation in x, scale in x, etc.) that can then be concatenated together to form a complex transform. The interface of a procedural transform is simple in that it takes a point as input and returns the transformed point as the output. The details of the transformation are hidden and the mechanics of the transformation can be performed by any desired method as long as the class conforms to the simple interface.

For this research, transforms for translation, rotation, and scale were developed. Because of the extensible framework, any other type of transformation can be easily added. Table 5 details the transformation implementation in the procedural transform as well as the corresponding homogeneous transform for the sake of clarity. In the following table, P is the original point and P' is the transformed point. A translation by a linear amount in the X, Y, or Z-direction is represented by $t_x, t_y,$ and t_z, respectively. A rotation in angular units about the X, Y, or Z-axes is represented by θ, ψ, and ϕ, respectively.

Table 5: List of procedural transforms and their corresponding homogeneous transform

Transform Name	Procedural Transform Implementation	Corresponding Homogenous Transform
X Translation	$P'_x = P_x + t_x$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
Y Translation	$P'_y = P_y + t_y$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
Z Translation	$P'_z = P_z + t_z$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
X Rotation	$P'_y = P_y \cos(\theta) - P_z \sin(\theta)$ $P_z' = P_y \sin(\theta) + P_z \cos(\theta)$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
Y Rotation	$P'_x = P_x \cos(\psi) + P_z \sin(\psi)$ $P_z' = -P_x \sin(\psi) + P_z \cos(\psi)$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 1 & & 0 \\ -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
Z Rotation	$P'_x = P_x \cos(\phi) - P_y \sin(\phi)$ $P_y' = P_x \sin(\phi) + P_y \cos(\phi)$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
X Scale	$P'_x = s_x P_x$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
Y Scale	$P'_y = s_y P_y$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$
Z Scale	$P_z' = s_z P_z$	$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$

As shown in the previous table, the procedural transform implementation does not require the use of matrix operations, thereby, reducing the number of mathematical operations needed to perform the transform. Though the form of the procedural transform does not appear as general as the homogenous transform, its implementation does meet the interface requirements of the operation, making the concatenation of multiple transforms both easy and efficient.

Point To Entity Deviations

Now that a method for transforming points to a point cloud has been formulated, a measure of point-to-model distance must be established. In the following sections, the mathematical formulations for point-to-entity calculations are derived for all of the geometry types addressed in this research.

It should be noted that the first three geometries discussed are typically considered two-dimensional in nature, and their corresponding derivations assume that the geometry and the point for the deviation calculation lie in a pre-specified plane. If three-dimensional calculations are needed for two-dimensional geometry, the methods derived in the last part of this section can be used, though with increased computational complexity. In real world applications, the data acquired for these geometries typically come from edge detection routines that only output points on a specified/measured plane; therefore, the derivations in the beginning of this section are both more applicable and more efficient.

Straight Line

The first of the geometries handled in this research is a straight line. A line is defined by a root point, $\vec{R}$, and a direction, $\vec{D}$. Figure 11 illustrates the definition of an infinite line. The distance from a point in space to the line then becomes:

$$\vec{V}_1 = \vec{P} - \vec{R} \tag{3}$$

$$\vec{V}_2 = (\vec{V}_1 \bullet \vec{D})\vec{D} \tag{4}$$

$$e = \left|\vec{V}_1 - \vec{V}_2\right| \tag{5}$$

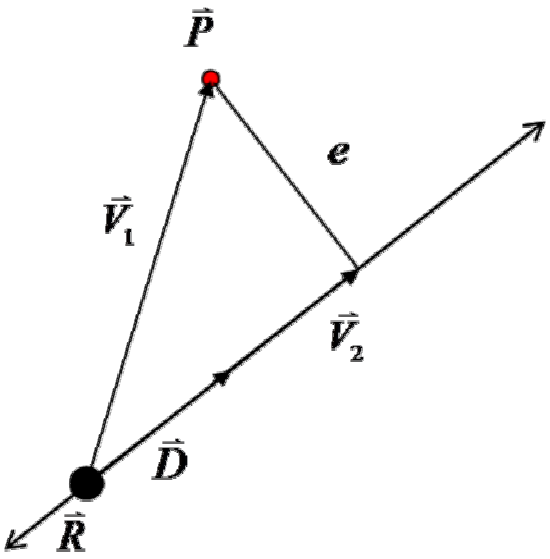

Figure 11: Straight line deviation calculation

Circular Curve

The second geometry considered in this research is a circular curve. A circular curve is defined by two parameters, a center point, $\vec{C}$, and a radius, R. Figure 12 illustrates the definition of a circular curve. For a circular curve, the deviation calculation is formulated as follows:

$$e = \left|\vec{P} - \vec{C}\right| - R \tag{6}$$

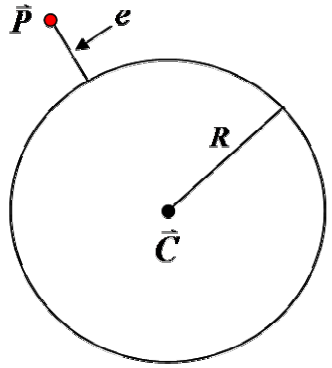

Figure 12: Circular curve deviation calculation

General Parametric Curve

General parametric curves are described by many geometric definitions, including but not limited to: Hermite, Bezier, and B-spline curves. B-spline curves have become the de facto standard in industry, particularly the nonuniform rational B-spline (NURBS). The point deviation equations for general parametric curves are not closed-form as are those for analytic curves previously discussed. The complex point deviation calculations for general parametric curves for this research are executed internally by the ACIS modeling kernel. The interface for this calculation fits within the programming constructs of the software platform developed in this research and, thus, is easy to

implement. In fact, any geometry not included in this research can be easily programmed into the framework, assuming the implementation of the geometry can calculate the normal distance to a point in question and, optionally, provide the first-order derivative of the calculation.

In most modeling kernels, the geometry of general curves is exclusively parametric. Though the exact method of determining the closest point to a parametric curve is not known, a general approach to the problem is presented here. Piegl and Tiller (1997) categorize this problem as a general point projection problem. One method to solve this problem is by using a bisection method. Usually, curves are parameterized on the interval of $u \in [0,1]$. One can start by evaluating the parametric curve at the two end points. Finding the distance between the point in question to these two endpoints then choosing the shorter of the distances, $u \in [0,1]$ is then subdivided into $u \in [0, 0.5]$ or $u \in [0.5,1]$ depending on which endpoint is closest to the point in space. The distance from the point in space to the curve at $u = .5$ is calculated. This distance is compared with the previous one, and the smaller of the two is taken. Again, the subinterval is divided and the process is repeated until the desired tolerance is met. A visual representation for this method is shown in Figure 13.

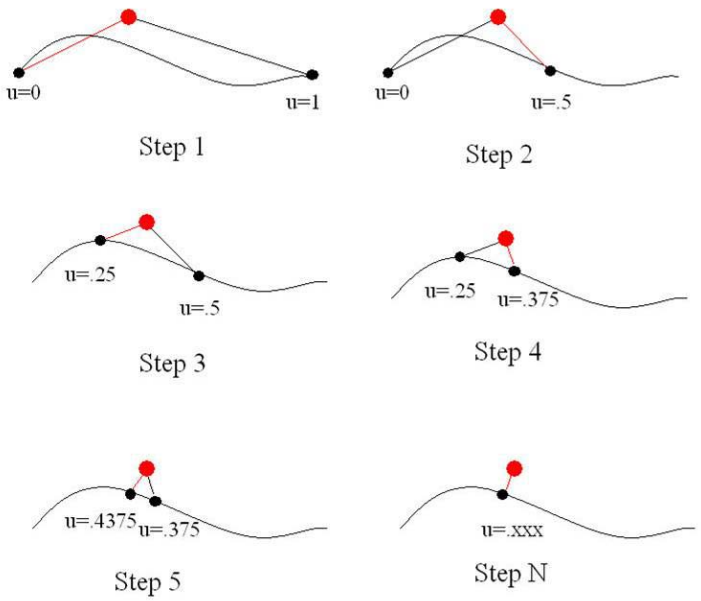

Figure 13: Simplified example of parametric deviation calculation

The previous example is a simplified case of solving the general point projection problem for a parametric curve. Other methods (e.g. Newton's method) can be used to find the closest point on a parametric curve which are both more efficient and reduce the number of singularities that can occur. Tucker (2000) goes through the laborious process of deriving analytical first-order information for NURBS, but implementation of the derivations was deemed impractical and unnecessary for the current research, especially since all current CAD models for LIGA parts consist of analytical geometry.

Plane

A plane is an extension of the line derivation in three-dimensional space. The simplest definition of a plane consists of a root point, $\vec{R}$, and a normal direction, $\hat{n}$. The deviation from a point to a plane is then:

$$e = \left|\vec{R} - \vec{P}\right| \bullet \hat{n} \tag{7}$$

Sphere

The sphere derivation is also just an extension of the circular curve in three-dimensional space. For a sphere, a center point, $\vec{P}$, and a radius, R, are required. The deviation equation is computed as follows:

$$e = \left|\vec{C} - \vec{P}\right| - R \tag{8}$$

Cylinder

A cylinder is defined by a point on the axis, $\vec{C}$, a normal direction, $\hat{n}$, representing the axis of the cylinder, and a radius, R. Given a point, $\vec{P}$, in three-dimensional space, the deviation calculation becomes:

$$\vec{V}_1 = \vec{P} - \vec{C} \tag{9}$$

$$\vec{V}_2 = (\vec{V}_1 \bullet \hat{n})\hat{n} \tag{10}$$

$$e = \left|\vec{V}_1 - \vec{V}_2\right| - R \tag{11}$$

Figure 14 illustrates the deviation calculation for a cylinder.

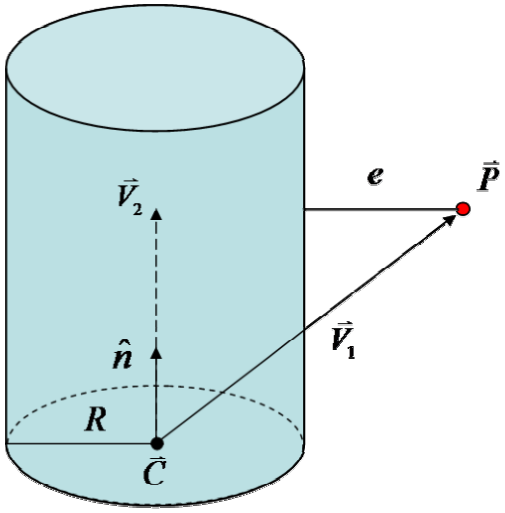

Figure 14: Cylinder deviation calculation

Cone

A cone is defined by a point on the axis, $\vec{C}$, the normal direction of the axis, $\vec{n}$, a radius, R, and a cone angle, θ. The deviation for a cone is calculated using Equations (12) - (17). Figure 15 illustrates the defined vectors for the calculations.

$$\vec{V}_1 = \vec{P} - \vec{C} \tag{12}$$

$$\vec{V}_2 = \vec{V}_1 - (\vec{V}_1 \bullet \hat{n})\hat{n} \tag{13}$$

$$\vec{V}_3 = \vec{V}_1 - \vec{V}_2 \tag{14}$$

$$\vec{V}_n = \cos(\theta)\frac{\vec{V}_3}{|V_3|} + \sin(\theta)\hat{n} \tag{15}$$

$$\vec{V}_4 = \vec{V}_1 - R\frac{\vec{V}_3}{|\vec{V}_3|} \tag{16}$$

$$e = \vec{V}_4 \bullet \frac{\vec{V}_n}{|\vec{V}_n|} \tag{17}$$

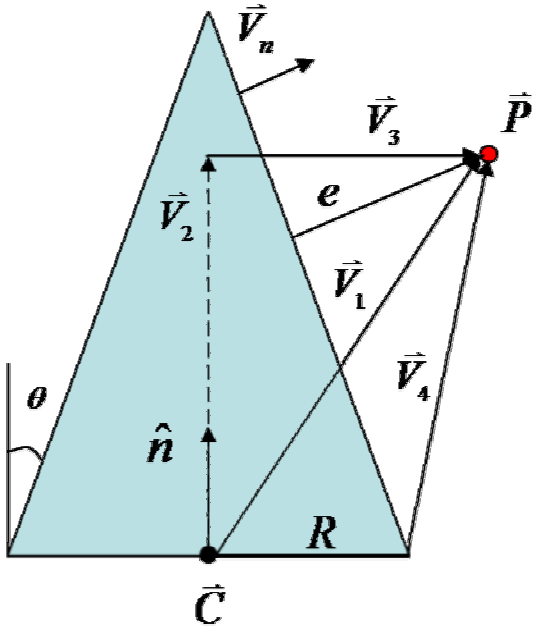

Figure 15: Cone deviation calculation

Torus

A torus is defined by the following: a center point, $\vec{C}$; a unit vector, $\hat{n}$; a major radius, R; and a minor radius, r. The steps involved in determining the deviation from a torus to a point in 3-D space are detailed in Equations (18) - (21). Figure 16 illustrates the deviation computation.

$$\vec{V}_1 = \vec{P} - \vec{C} \tag{18}$$

$$\vec{V}_2 = \vec{V}_1 - (\vec{V}_1 \bullet \hat{n})\hat{n} \tag{19}$$

$$\vec{V}_3 = \vec{V}_1 - R\frac{\vec{V}_2}{\left|\vec{V}_2\right|} \tag{20}$$

$$e = \left|\vec{V}_3\right| - r \tag{21}$$

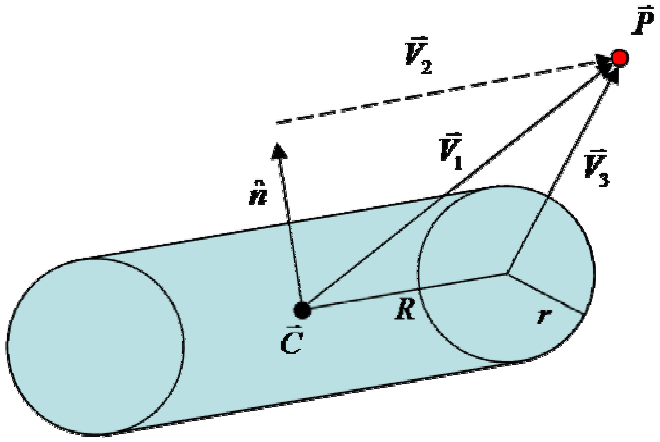

Figure 16: Torus deviation calculation

General Parametric Surface

A general parametric surface is just an extension of the general parametric curve previously discussed. Again, the geometric modeler is responsible for providing the deviation calculation of any surface that cannot be analytically represented by the geometries previously described.

3-D Deviations for 2-D Geometry and Trimmed Geometry

The previous derivations for 2-D geometry assumed that the data associated with the geometry are co-planar. For the applications of this research, this assumption typically holds. The majority of the data associated with the 2-D CAD models is

generated from vision systems using edge detection algorithms. The majority of these algorithms output data only in two dimensions. The vision system itself may append a corresponding z-height to the data based on a previous auto focus routine, but the data are still planar. In order to accommodate future needs, derivations for 3-D deviations to 2-D geometry are discussed here. For general parametric geometry, the 3-D deviation is calculated with the modeling kernel and is not discussed.

Straight line

The deviation of a point to a line remains the same as the previous derivation. The point on the line and its corresponding direction are extended to hold information for the third dimension.

Circular Curve

In order to compute the 3-D distance from a point to a circular curve, one extra parameter must be defined for the circle. This parameter is a normal vector, $\hat{n}$, which is perpendicular to the plane in which the circle lies. The definition of a 3-D circle is essentially a torus with a minor radius of zero. Having this new definition of a circular curve, the deviation from a point in 3-D space can be computed using Equations (22) - (25). Figure 17 illustrates this deviation computation.

$$\vec{V}_1 = \vec{P} - \vec{C} \tag{22}$$

$$\vec{V}_2 = \vec{V}_1 - (\vec{V}_1 \bullet \hat{n})\hat{n} \tag{23}$$

$$\vec{V}_3 = R\frac{\vec{V}_2}{\left|\hat{V}_2\right|} \tag{24}$$

$$e = \left| \vec{V}_1 - \vec{V}_3 \right| \tag{25}$$

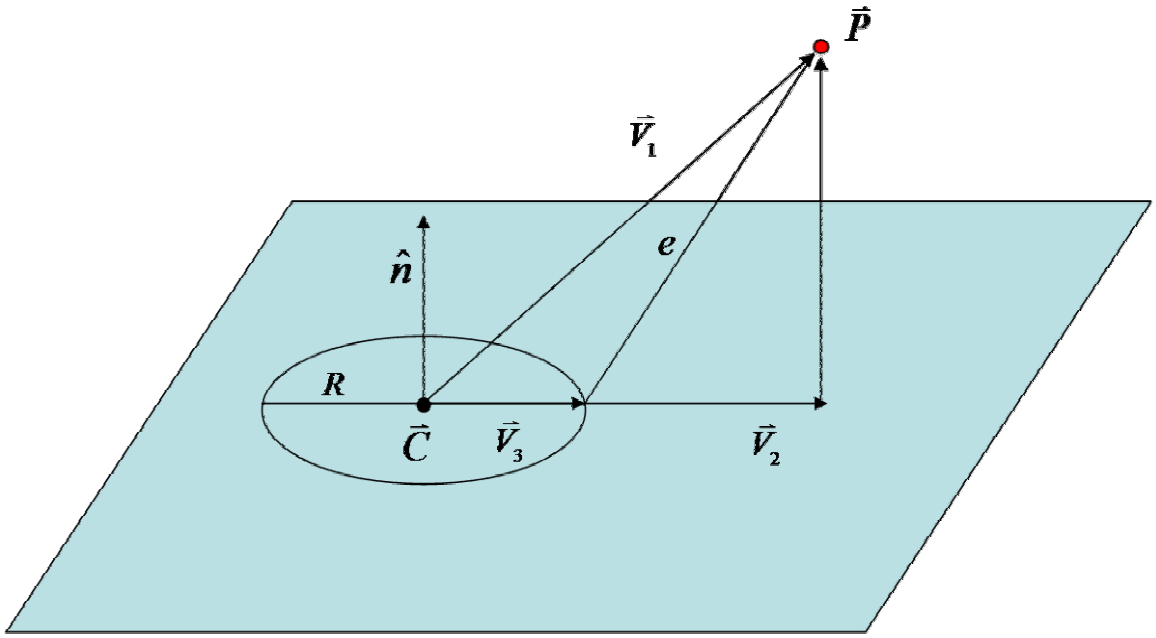

Figure 17: 3-D deviation calculation to a circular curve

Trimmed Geometry

All of the previous deviation calculations use the full geometric definition of an entity. In modern CAE applications, the curve and surface definitions are typically different from their corresponding edge and face definitions, respectively. This permits an edge or face to be trimmed (e.g. quarter of a sphere) while maintaining the full geometric definition. A problem can arise if only a partial edge or face exists, but a deviation calculation is computed using the full geometric definition of that entity. Figure 18 illustrates a CAD model which is a quarter of a unit circle. The CAD geometry, however, holds the entire geometric definition for the circle and, thus, an incorrect deviation calculation is computed.

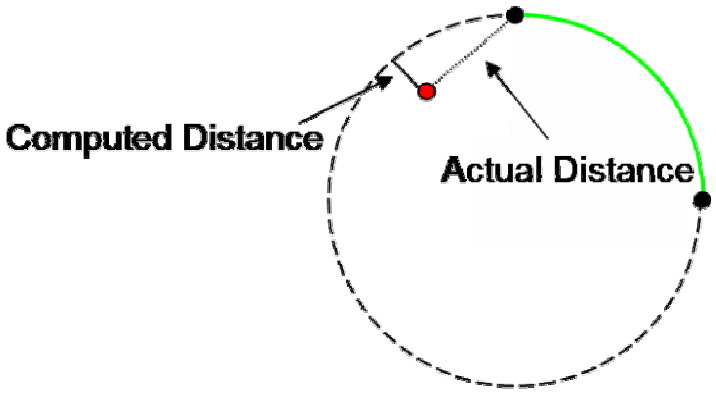

Figure 18: Illustration of trimmed geometry

Determining the deviation from a point to a trimmed edge or face adds significantly to the complexity of the deviation computation. In all practicality, determining the deviation to a trimmed edge or face is unnecessary, since the point cloud typically is positioned close enough to the entity so that the correct deviation is calculated. In order to provide a level of robustness, the following derivations can be incorporated in order to handle these special cases.

Straight Line

To calculate the deviation from a point to a line segment, the previous definition of a line must be modified. The line segment must be parameterized in order to incorporate the two endpoints. Typically, the parameterization value varies in the range from 0 to 1. The root point, $\vec{R}$, must lie at one of the endpoints, and the direction, $\vec{D}$, must be representative of the length of the line segment. The new definition of the line is shown in Equation (26).

$$\vec{L}(u) = \vec{R} + \vec{D}u \qquad u \in [0,1] \tag{26}$$

In order to calculate the deviation from a point, $\vec{P}$, to the line, the closest point on the line to $\vec{P}$ must be determined. The closest point on the line is the projection of $\vec{P}$ onto the line as shown in Equation (27).

$$\vec{P}_{proj} = \vec{R} + \vec{D}u_0 \quad \text{where} \tag{27}$$

$$u_0 = \frac{(\vec{P} - \vec{R}) \bullet \vec{D}}{\vec{D} \bullet \vec{D}} \tag{28}$$

The deviation to the line segment from a point then becomes:

$$e = \begin{Bmatrix} \left|\vec{P} - \vec{R}\right| & u_0 < 0 \\ \left|\vec{P} - \vec{P}_{proj}\right| & 0 \le u_0 \le 1 \\ \left|\vec{P} - (\vec{R} + \vec{D})\right| & u_0 > 1 \end{Bmatrix} \tag{29}$$

Circular Curve

The circular curve definition must also be redefined in order to address trimmed edges. The redefinition of the circle includes a center, $\vec{C}$, a start point, $\vec{S}$, an end point, $\vec{E}$, and a mid-point, $\vec{M}$. The derivatives at each point, excluding the center point, must also be defined. Figure 19 illustrates the redefinition for a partial circular curve.

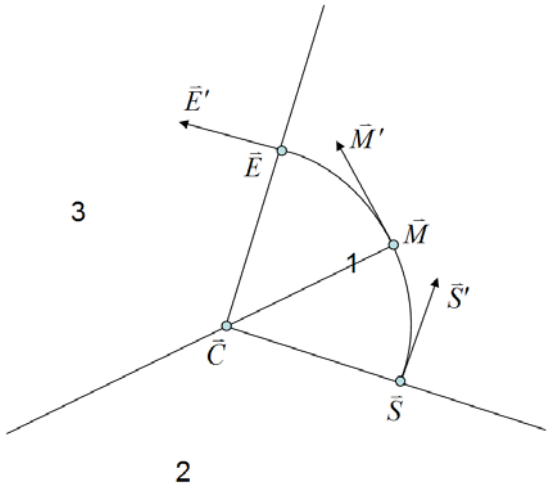

Figure 19: Definition for trimmed circular curve

From the above figure, the space around a circular arc can be partitioned into three areas. Any point that lies in area 1 falls into the traditional deviation calculation previously derived. Any point that lies in area 2 is closest to the starting point of the arc, $\vec{S}$, and the deviation should be calculated to that point. Any point that lies in area 3 is closest to the end point of the arc, $\vec{E}$, and the deviation should be calculated to that point. In order to efficiently determine in which area a given point lies, a mid-point, $\vec{M}$, is needed as well as the corresponding first derivatives of each point of the circular arc. Having these definitions and a given point in space, $\vec{P}$, the deviation to a trimmed circle can be derived as follows:

$$\vec{V}_1 = \vec{P} - \vec{C} \tag{30}$$

$$\begin{aligned} &\text{if } (\vec{V}_1 \bullet \vec{E}') < 0 \text{ and } (\vec{V}_1 \bullet \vec{S}') > 0 \\ &\text{then } e = \left|\vec{P} - \vec{C}\right| - R \end{aligned} \tag{31}$$

$$\text{if } (\vec{V}_1 \bullet \vec{E}') > 0 \text{ and } (\vec{V}_1 \bullet \vec{M}') > 0 \quad \text{then } e = \left|\vec{P} - \vec{E}\right| \tag{32}$$

$$\text{if } (\vec{V}_1 \bullet \vec{E}') < 0 \text{ and } (\vec{V}_1 \bullet \vec{M}') < 0 \quad \text{then } e = \left|\vec{P} - \vec{S}\right| \tag{33}$$

General Parametric Curves

In this research, the complex algorithms for deviation calculations for general parametric curves are left to the modeling kernel, and, accordingly, determining if the closest point actually lies on the edge is left to the modeling kernel.

Three-dimensional Geometry

Working with trimmed surfaces is a field unto itself and, thus, derivations for these situations are beyond the scope of this work. A possible solution to this problem is to use the modeling kernel to test whether the calculated closest point lies on the surface of the given CAD model. This functionality is provided as an option to the user due to the increased computation times of the point-in-face test. Another method would be to facet the CAD model and use a proven point-in-polygon test. Again, for most real world applications, these tests should be unnecessary but are noted here for special cases.

Point-to-Entity Assignments

All of the deviations calculated up to this point have been for one specific type of geometry and only one instance of that geometry. With most CAD models, multiple

edges and faces are present with dissimilar geometries. Therefore, before the actual deviation calculations are computed, each point must be assigned to a specified geometry.

Two approaches for point-to-entity assignment are implemented in this research. The most obvious and programmatically elegant method is to compare each point to each face and determine the closest entity to that point. For a CAD model with a small number of entities, this method works well and is simple to implement. Current high aspect ratio MEMS designs typically consist of tens or hundreds of elements, and an exhaustive comparison for point assignment is not overly time consuming. Table 6 lists the pseudo-code for an implementation of an exhaustive search between a point cloud and a CAD model.

Table 6 : Pseudo-code for exhaustive search algorithm

```
for (i = 1 : i = num_pts)
   for(j = 1 : j = num_entities)
      e = deviation from point(i) to entity(j)
      if (e < global_min)
      {
         global_min = e
         closest_entity(i) = entity(j)
      }
   end
end
```

As more complex designs become feasible for MEMS manufacturing and the number of entities approaches the thousands, another scheme for point assignment may be necessary. Claudet (2001) proposes a sweep-volume approach to point assignment for general 3-D CAD models. This research extends this model to include both two-dimensional and three-dimensional geometry (i.e. sweep lines and sweep planes).

The idea of sweep lines and planes is not new in the field of computational geometry for tasks such as sorting and visualization (Fortune 1997). Using a sweep

method requires the use of a priority queue to hold certain "events." In the case of point-to-entity assignments, three events are used in the queue: upper bounding box events, lower bounding box events, and point events. The bounding boxes used in a sweep assignment algorithm are typically minimum bounding boxes of each geometric entity in the CAD model, expanded by some factor, ε. This expansion factor can be determined from either a certain percentage of a characteristic dimension of the part or from *a priori* knowledge of the expected maximum deviation of the inspected data. Once the bounding boxes have been established, the sweep algorithm can begin to process events. Conceptually, this step can be seen as a line or plane sweeping from $-\infty$ to ∞ along one of the principal axes. When the line or plane encounters the "top" of a bounding box, the geometry belonging to that bounding box is added to the queue. When a point is encountered, a point event is generated and the deviations from that point to every geometric entity corresponding to each bounding box in the queue are calculated. The smallest deviation is recorded and that point is assigned to the closest entity in the queue. When the line or plane encounters the "bottom" of a bounding box, the corresponding geometry is removed from the queue. Once the line or plane has swept through the scene, every point is assigned and the corresponding minimum deviation is recorded. This method has the advantage of not having to compare each point to each entity in the CAD model. Only entities that are relatively close to the given point are used, and a significant performance gain can be realized for models with many faces or edges. Claudet (2001) is referenced for performance characteristics of exhaustive and volume-sweep methods.

It should also be noted that for the sweep method implemented in this research, point-to-entity assignments are calculated at each iteration of the registration process. It is feasible to further reduce computation time if a method to limit the number of point-to-entity assignments is implemented (e.g. trust-region assignment). These types of methods, however, have the drawback of possibly assigning a point to the incorrect geometric entity. In order to avoid such errors, this research implements a sweep method which reassigns point data during each iteration of the registration process.

Registration

Having derived methods to transform a point cloud, compute deviations from the point cloud to the nominal geometry, and assign points to individual entities in the CAD model, a method to minimize the point deviations to the model is developed. As previously mentioned, the problem of registering the data to the CAD model can be posed as a nonlinear least-squares problem which is defined as follows:

$$\begin{aligned} &\text{Given } R:\mathbb{R}^n \to \mathbb{R}^m,\ m \geq n, \\ &\text{find } \quad x_* \in \mathbb{R}^n \text{ for which } \sum_{i=1}^{m} (r_i(x))^2 \text{ is minimized} \end{aligned} \tag{34}$$

This problem is frequently encountered in curve fitting, and the choice for the sum-of-squares measure is justified by statistical considerations (Bard 1970). For this application, the residual functions, $r_i(x)$, are the deviation calculations derived previously and x is the vector of variables typically representing the transformation parameters.

Minimization

Solutions to the problem of nonlinear least-squares have been studied for over thirty years, and many consider the field mature. The majority of algorithms developed to solve this problem are derived from a prototype of Newton's method. For non-linear functions, the general idea is to create a simplified model, m_c, which is typically quadratic, and solve for the root of that model at each iteration. The algorithm chosen for minimization in this research is based on the work of Levenberg (1944) and Marquardt (1963). Before this method can be presented, derivations of the Newton and Gauss-Newton methods are required.

The least-squares problem can be defined more succinctly as,

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x)=\frac{1}{2}R(x)^T R(x)=\frac{1}{2}\sum_{i=1}^{m} r_i(x)^2 \tag{35}$$

The first derivative of (35) can be derived as,

$$\nabla f(x)=\sum_{i=1}^{m} r_i(x)\cdot\nabla r_i(x)=J(x)^T R(x) \tag{36}$$

Similarly, the second derivative of (35) is

$$\begin{aligned}\nabla^2 f(x)&=\sum_{i=1}^{m}(\nabla r_i(x)\cdot\nabla r_i(x)^T+r_i(x)\cdot\nabla^2 r_i(x))\\&=J(x)^T J(x)+S(x)\end{aligned} \tag{37}$$

where

$$S(x) \triangleq \sum_{i=1}^{m} r_i(x) \cdot \nabla^2 r_i(x) \tag{38}$$

Using a Taylor series quadratic model around the a point, x_c, the model equation becomes

$$\begin{aligned} m_c(x) &= f(x_c) + \nabla f(x_c)^T (x - x_c) + \frac{1}{2}(x - x_c)^T \nabla^2 f(x_c)(x - x_c) \\ &= \frac{1}{2} R(x_c)^T R(x_c) + R(x_c)^T J(x_c)(x - x_c) \\ &\quad + \frac{1}{2}(x - x_c)^T (J(x_c)^T J(x_c) + S(x_c))(x - x_c) \end{aligned} \tag{39}$$

Applying Newton's method to equation (35), the Newton step becomes

$$x_+ = x_c - (J(x_c)^T J(x_c) + S(x_c))^{-1} J(x_c)^T R(x_c) \tag{40}$$

The previous derivation of Newton's method is q-quadratically convergent. The major problem with this method, though, is that $S(x)$ is usually unavailable or too expensive to calculate. The Gauss-Newton method simplifies the problem by the use of an affine model as shown in Equation (41).

$$m_c(x) = R(x_c) - J(x_c)(x - x_c) \tag{41}$$

The least-squares problem is then defined as

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \left\| R(x_c) - J(x_c)(x - x_c) \right\|_2^2 \tag{42}$$

The Gauss-Newton step then becomes

$$x_+ = x_c - (J(x_c)^T J(x_c))^{-1} J(x_c)^T R(x_c) \tag{43}$$

The difference between the Newton and Gauss-Newton method is the omission of the $S(x)$ term in the Gauss-Newton method. The Gauss-Newton method is only q-quadratically convergent if $S(x_*)$ is zero, meaning that we have a zero-residual problem (i.e. the data exactly match the nominal geometry); otherwise, the algorithm is only q-linearly convergent. Levenberg and Marquardt proposed the modification of using a trust region to augment the Gauss-Newton method. The least-squares problem using a trust region is redefined as follows:

$$\begin{aligned} &\underset{x_+ \in \mathbb{R}^n}{\text{minimize}} \quad \left\| R(x_c) - J(x_c)(x_+ - x_c) \right\|_2 \\ &\text{subject to} \qquad \left\| x_+ - x_c \right\|_2 \le \delta_c \end{aligned} \tag{44}$$

The Levenberg-Marquardt step is then defined as

$$x_+ = x_c - (J(x_c)^T J(x_c) - \mu_c I)^{-1} J(x_c) R(x_c) \tag{45}$$

where $\mu_c = 0$ if $\delta_c \ge \left\| (J(x_c)^T J(x_c))^{-1} J(x_c) R(x_c) \right\|_2$ and $\mu_c > 0$ otherwise. For methods on choosing appropriate values for μ_c and δ_c, Moré (1977) is referenced. The convergence rate of the Levenberg-Marquardt method is in between that of the Newton method and the Gauss-Newton method. For problems in which the residual vector is relatively small, the Levenberg-Marquardt method is q-quadratically convergent. For a

typical CAI scenario, this assumption is often reasonable. In practice, the Levenberg-Marquardt has been very successful, and in this research an adaptation of the Fortran solver found in MINPACK (Moré 1980) is developed.

First-Order Information

It should be noted that there is the requirement of first-order information from the Jacobian in all of the previous derivations. There are two approaches to providing this Jacobian information: analytical and numerical derivations.

For general parametric curves and surfaces, the modeling kernel is responsible for handling the analytic derivations of first-order information. This requirement follows from the same reasons as the deviation calculations for general parametric entities. Claudet (2001) provides the mathematical framework for the analytical computations of the Jacobian for all analytical 3-D shapes and the procedural transforms. This framework extends those formulations for the 2-D analytical geometry analyzed in this research. The first-order information is geometry and transform specific, and the details of the formulations are shown in the following sections. In order to calculate the partials with respect to the geometric entities presented in this work, some vector notation should be presented to clarify the operations in the following sections. Namely, the partial of a vector's magnitude is

$$\frac{\partial \left|\vec{V}\right|}{\partial \vec{\xi}} = \frac{\vec{V}}{\left|\vec{V}\right|} \bullet \frac{\partial \vec{V}}{\partial \vec{\xi}} = \hat{v} \bullet \frac{\partial \vec{V}}{\partial \vec{\xi}} \tag{46}$$

and the partial of a normalized vector is

$$\frac{\partial \hat{v}}{\partial \vec{\xi}} = \frac{\partial \frac{\vec{V}}{|\vec{V}|}}{\partial \vec{\xi}} = \frac{1}{|\vec{V}|}\frac{\partial \vec{V}}{\partial \vec{\xi}} - \frac{1}{|\vec{V}|^3}\vec{V} \bullet \left(\vec{V} \bullet \frac{\partial \vec{V}}{\partial \vec{\xi}} \right) \tag{47}$$

Procedural Transforms

First-order information for the transforms presented in this work is readily derived. Table 7 displays the first-order information for the transformations used in this research. First-order information for any type of transformation is easily added to the current framework because of the extensible nature of developed algorithms.

Table 7 : First-order information of procedural transfoms

Transform Name	First-Order Vector Information
X Translation	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial t_x}\right] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$
Y Translation	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial t_y}\right] = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
Z Translation	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial t_z}\right] = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
X Rotation	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial \theta}\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin(\theta) & -\cos(\theta) \\ 0 & \cos(\theta) & -\sin(\theta) \end{bmatrix} \vec{P}$
Y Rotation	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial \psi}\right] = \begin{bmatrix} -\sin(\psi) & 0 & \cos(\psi) \\ 0 & 0 & 0 \\ -\cos(\psi) & 0 & -\sin(\psi) \end{bmatrix} \vec{P}$
Z Rotation	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial \phi}\right] = \begin{bmatrix} -\sin(\phi) & -\cos(\phi) & 0 \\ \cos(\phi) & -\sin(\phi) & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{P}$
X Scale	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial s_x}\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{P}$
Y Scale	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial s_y}\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{P}$
Z Scale	$\frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} = \left[\frac{\partial \vec{P}^{(1)}}{\partial s_x}\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{P}$

The first-order information for procedural transforms is just the first set of calculations needed to calculate the first-order information of the entire deviation

calculation from a given point to a specific geometric entity. The first-order information with respect to the specific geometry deviation calculation is still needed. Breaking up the partial calculations allows for a modular framework that can easily be extended for future needs.

Straight Line

The deviation from a straight line to a point in space was derived with Equations (3) - (5). The partials with respect to a given set of transformation parameters can be calculated using equations

$$\frac{\partial \vec{V}_1}{\partial \vec{\xi}} = \frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} \tag{48}$$

$$\frac{\partial \vec{V}_2}{\partial \vec{\xi}} = \hat{n} \bullet \frac{\partial \vec{V}_1}{\partial \vec{\xi}} \tag{49}$$

$$\frac{\partial e}{\partial \vec{\xi}} = \frac{\vec{V}_1 - \vec{V}_2}{\left| \vec{V}_1 - \vec{V}_2 \right|} \bullet \left(\frac{\partial \vec{V}_1}{\partial \vec{\xi}} - \frac{\partial \vec{V}_2}{\partial \vec{\xi}} \right) \tag{50}$$

Circular Curve

The deviation calculation for a circular curve was derived in Equation (6). Applying property (46), the first-order information for this geometric definition is derived as follows:

$$\frac{\partial e}{\partial \vec{\xi}} = \frac{\vec{P} - \vec{C}}{\left| \vec{P} - \vec{C} \right|} \bullet \frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} \tag{51}$$

Plane

Using the deviation equation shown in Equation (7), the partial of the deviation equation with respect to a set of procedural transform parameters is calculated in (52).

$$\frac{\partial e}{\partial \vec{\xi}} = \hat{n} \bullet \frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} \tag{52}$$

Sphere

Using property (46) and applying it to the deviation equation of a sphere previously derived in Equation (8), the first-order information of the deviation equation becomes

$$\frac{\partial e}{\partial \vec{\xi}} = \frac{\vec{P} - \vec{C}}{\left| \vec{P} - \vec{C} \right|} \bullet \frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} \tag{53}$$

Cylinder

From the derivation of the cylinder error deviation shown in Equations (9) - (11), the partials can be computed using Equations (54) - (56).

$$\frac{\partial \vec{V}_1}{\partial \vec{\xi}} = \frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} \tag{54}$$

$$\frac{\partial \vec{V}_2}{\partial \vec{\xi}} = \hat{n} \bullet \frac{\partial \vec{V}_1}{\partial \vec{\xi}} \tag{55}$$

$$\frac{\partial e}{\partial \vec{\xi}} = \frac{\vec{V}_1 - \vec{V}_2}{\left| \vec{V}_1 - \vec{V}_2 \right|} \bullet \left(\frac{\partial \vec{V}_1}{\partial \vec{\xi}} - \frac{\partial \vec{V}_2}{\partial \vec{\xi}} \right) \tag{56}$$

Cone

The deviation calculation for a cone is derived in Equations (12) - (17). The partial with respect to a vector of transform parameters are calculated as follows:

$$\frac{\partial \vec{V}_1}{\partial \vec{\xi}} = \frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} \tag{57}$$

$$\frac{\partial \vec{V}_2}{\partial \vec{\xi}} = \hat{n}\left(\hat{n} \bullet \frac{\partial \vec{V}_1}{\partial \vec{\xi}}\right) \tag{58}$$

$$\frac{\partial \vec{V}_3}{\partial \vec{\xi}} = \frac{\partial \vec{V}_1}{\partial \vec{\xi}} - \frac{\partial \vec{V}_2}{\partial \vec{\xi}} \tag{59}$$

$$\frac{\partial \vec{V}_n}{\partial \vec{\xi}} = \cos\gamma * \left(\frac{1}{\left|\vec{V}_3\right|} * \frac{\partial \vec{V}_3}{\partial \vec{\xi}} - \frac{1}{\left|\vec{V}_3\right|^3} * \vec{V}_3 \bullet \left(\vec{V}_3 \bullet \frac{\partial \vec{V}_3}{\partial \vec{\xi}}\right)\right) \tag{60}$$

$$\frac{\partial \vec{V}_4}{\partial \vec{\xi}} = \frac{\partial \vec{V}_1}{\partial \vec{\xi}} - \frac{1}{\left|\vec{V}_3\right|}\frac{\partial \vec{V}_3}{\partial \vec{\xi}} - \frac{1}{\left|\vec{V}_3\right|^3} V_3 \bullet \left(\vec{V}_3 \bullet \frac{\partial \vec{V}_3}{\partial \vec{\xi}}\right) \tag{61}$$

$$\frac{\partial e}{\partial \vec{\xi}} = \vec{V}_4 \bullet \left(\frac{1}{\left|\vec{V}_n\right|} * \frac{\partial \vec{V}_n}{\partial \vec{\xi}} - \frac{1}{\left|\vec{V}_n\right|^3} * \vec{V}_n \bullet \left(\vec{V}_n \bullet \frac{\partial \vec{V}_n}{\partial \vec{\xi}}\right)\right) + \frac{\vec{V}_n}{\left|\vec{V}_n\right|} \bullet \frac{\partial \vec{V}_4}{\partial \vec{\xi}} \tag{62}$$

Torus

The calculation of the error deviation for a torus is shown in Equations (18) - (21). Calculating the partial with respect to transformations is shown in Equations (63) - (66).

$$\frac{\partial \vec{V}_1}{\partial \vec{\xi}} = \frac{\partial \vec{P}^{(1)}}{\partial \vec{\xi}} \tag{63}$$

$$\frac{\partial \vec{V}_2}{\partial \vec{\xi}} = \frac{\partial \vec{V}_1}{\partial \vec{\xi}} - \hat{n}\left(\hat{n} \bullet \frac{\partial \vec{V}_1}{\partial \vec{\xi}}\right) \tag{64}$$

$$\frac{\partial \vec{V}_3}{\partial \vec{\xi}} = \frac{\partial \vec{V}_1}{\partial \vec{\xi}} - R_1\left(\frac{1}{\left|\vec{V}_2\right|}\frac{\partial \vec{V}_2}{\partial \vec{\xi}} - \frac{1}{\left|\vec{V}_2\right|^3}\vec{V}_2\left(\vec{V}_2 \bullet \frac{\partial \vec{V}_2}{\partial \vec{\xi}}\right)\right) \tag{65}$$

$$\frac{\partial e}{\partial \vec{\xi}} = \frac{\vec{V}_3}{\left|\vec{V}_3\right|} \bullet \frac{\partial \vec{V}_3}{\partial \vec{\xi}} \tag{66}$$

In order to provide a higher level of robustness, this research also implements a numerical algorithm to calculate the Jacobian. This feature provides a less restrictive requirement on the objective functions (i.e. deviation calculations) derived previously. The only requirement is that the deviation from the nominal geometry to the point in question be returned; no analytical calculation of first-order information is needed. Providing this functionality makes it much easier to add any type of geometry to the framework as long as the deviation to the geometry can be formulated. First-order information, particularly for spline geometry, can be numerically complex (Tucker 2000) and not always feasible to implement. Dennis and Schnabel (1996) show that for properly chosen finite-differences, the use of analytical and numerical derivatives are virtually indistinguishable for many problems. The numerical algorithm chosen for this research to numerically compute the Jacobian is a forward-difference formula. This method is chosen because of its proven robustness in practice in numerical libraries such as the Fortran MINPACK. The forward-difference formula for one element of the Jacobian is shown in equation (67).

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} \tag{67}$$

The step, h, is chosen to be a small multiple of the machine epsilon.

Summary

Having a method to spatially transform data, a method to calculate deviations to geometric entities, a method to assign points to individual entities, and a method to perform least-squares minimization, a framework is established to connect these algorithms and perform cloud-to-CAD registration. Figure 20 illustrates the flow of data in the registration process.

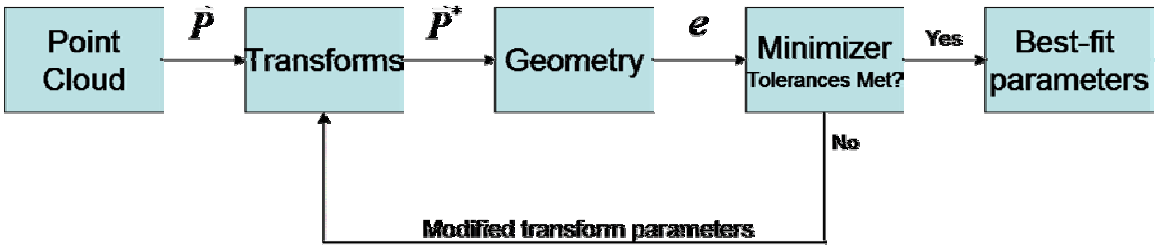

Figure 20: Data flow in registration process

MEMS Inspection and Fiducial Analysis

The previous sections dealt with the general analysis of point cloud data and CAD models. Using least-squares fitting to register 2-D or 3-D data can be applied to any manufactured product using data from any metrology system. The previously derived algorithms provide a level of extensibility that makes this research applicable to many different fields. The remainder of this chapter presents methods developed specifically for the inspection of high aspect ratio MEMS. The derivations in this section serve as the

main contribution of this research that fills an informational gap that currently exists in high aspect ratio MEMS metrology. This informational gap, namely, is the characterization of gross geometric errors in a part. Currently, only form errors can be characterized.

Before specific methods developed for MEMS inspection are discussed, a brief explanation of different types of part errors is in order. Part errors can generally be classified into three categories based on the scale of the errors. Surface roughness is typically the smallest error measured and is characterized by the shortest wavelength deviations of a part's surface. The next type of error typically measured is form error which is characterized by longer wavelength deviations of the part's surface. Form errors are typically associated with single surfaces or edges of a part and describe the deviation from the nominal geometry. Geometric errors are the largest errors in a part, typically characterizing errors between multiple surfaces or edges in a part.

Most CAI techniques are concerned with form errors in a part. In macro-scale parts, a feature can typically be inspected with one fixture setup. If more than one fixturing scheme is required, then known datums are typically incorporated in order to relate the multiple inspection routines. The traditional output from a macro-scale inspection is a three-dimensional point cloud representing all of the inspected surfaces on a part. When a full three-dimensional point cloud is available, traditional CAI techniques can be used to fully qualify a part. With micro-scale parts, full three-dimensional point clouds are currently unattainable. This limitation makes for difficult part verification.

This research proposes that some of the limitations of two-dimensional inspection of micro-scale parts can be overcome with the use of fiducial information. Fiducial

information can be defined as any measurable feature that is used for alignment purposes. Fiducial marks are commonly used in machine vision applications when image processing software is used for electronics inspection. In machine vision, fiducial marks serve as a method to provide some type of coordinate frame reference of which other features are inspected with respect to. This research proposes that fiducial marks which can be directly measured can be used to provide a common reference for multiple point cloud analysis.

A methodology has been developed to inspect multiple edges of a high aspect ratio MEMS device and relate the corresponding data. Currently, planar (i.e. two-dimensional) scans of the tops and bottoms of high aspect ratio parts are easily acquired with available metrology hardware. These scans can be used to determine form errors of the manufactured part by comparing the edge data to the nominal CAD geometry. For example, if a rectangular part is inspected, the top and bottom of the part can be inspected independently. Using a least-squares registration method, the top scan or the bottom scan can be compared to an ideal rectangle. For form errors, this type of analysis is fine, but no information can be deduced about the relationship between the two scans. If, however, a set of fiducial marks is available, it is possible to geometrically relate these two scans and infer some three-dimensional information about the part. The main benefit of this type of analysis is that gross geometric errors can be deduced to provide a quantitative analysis of the entire MEMS device.

The software framework developed in this work allows for the analysis of multiple point clouds in one scene (i.e. multiple point clouds can be compared to a single CAD model). For a MEMS device, it is feasible that a top and bottom scan of the edge

data can be imported into the platform and a least-squares registration applied to each cloud independently. If fiducial information is available, it is also possible to calculate some information about the geometric errors in the part. With fiducial information, geometric errors such as shift or twist in the part can be detected. Even if the least-squares analysis computes an acceptable form tolerance, the fiducial analysis may compute large geometric errors (i.e. shift and twist) between the two scans of data. For planar point clouds, the requirements for the fiducial analysis are the location of four points, two for the top scan and two for the bottom scan. A scenario of a rectangular block CAD model and two point clouds is depicted in Figure 21.

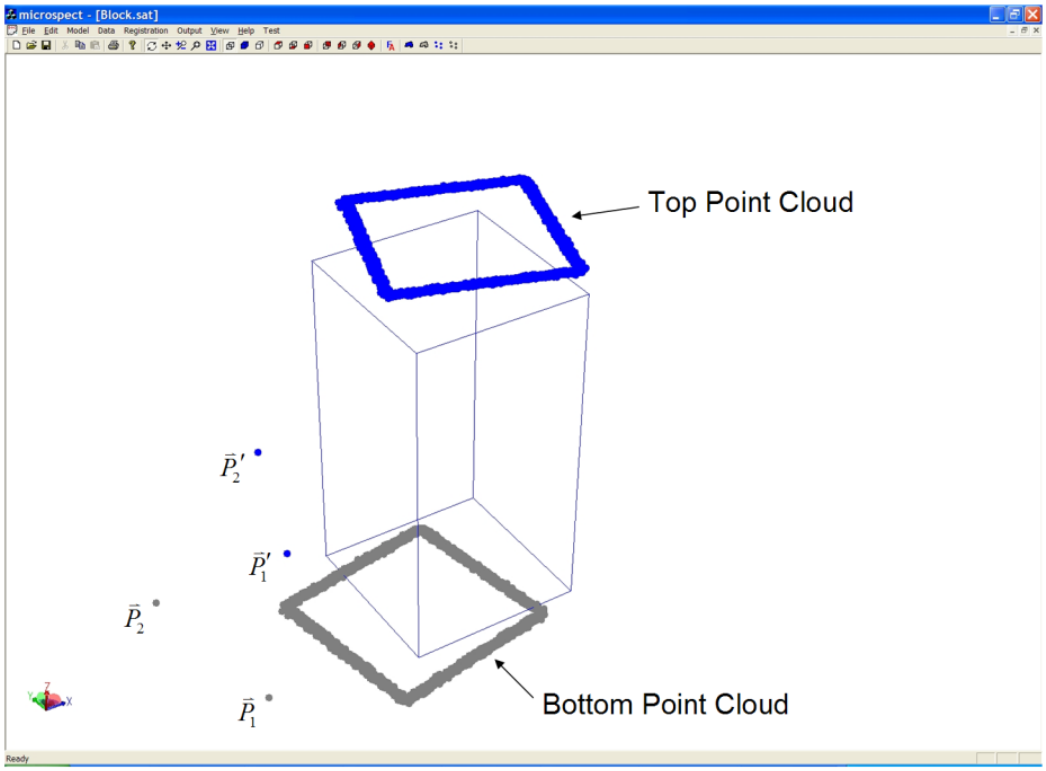

Figure 21: Fiducial points for multiple point clouds

In the above figure, two independently acquired point clouds are shown. Each point cloud contains fiducial information represented by two points in space, which are labeled in the figure. These two fiducial points are common to both point clouds and, thus, should be coincident.

The problem to be solved here is the calculation of shift and twist between the two point clouds. Shift errors can be defined as the translational error between two point clouds. For two-dimensional point clouds, translations in the X and Y-direction are of particular interest. For a high aspect ratio part, a shift error can be visualized as a cross-section which is not extruded orthogonal to the plane of the cross-section. A twist error can be defined as the rotation of a cross-sectional geometry along its axis of extrusion. When twist is being calculated, it is of extreme importance that an appropriate rotation point be specified for the calculations; otherwise, a shift will be computed that is not representative of the part. The rotation point that is typically be chosen for high aspect ratio parts is any point along the axis of extrusion of the part. Having corresponding fiducial points in multiple point clouds allows for the closed-form calculation of geometric errors (e.g. shift and twist) of a fabricated part. These geometric errors can be calculated *after* both point clouds are least-squares registered to the nominal geometry.

Letting $\vec{P}_1$ and $\vec{P}_2$ be the fiducial points for the first point cloud, and $\vec{P}_1'$ and $\vec{P}_2'$ being in the second point cloud, two vectors can be defined as follows:

$$\vec{V}_1 = \vec{P}_2 - \vec{P}_1 \tag{68}$$

$$\vec{V}_1' = \vec{P}_2' - \vec{P}_1' \tag{69}$$

The twist in the part using these two vectors can be calculated as

$$\theta = \cos^{-1}\left[\frac{(\vec{V}_1 \bullet \vec{V}_1')}{\left|\vec{V}_1\right|\left|\vec{V}_1'\right|}\right] \tag{70}$$

In order to determine any shift in the part, one has to take care when incorporating a rotation point, if necessary. If it can be assumed that there is no rotation between the data sets, the shift between the two data sets is simply $\vec{P}_1' - \vec{P}_1$ or $\vec{P}_2' - \vec{P}_2$. If there is the possibility of rotation between the data sets, a rotation point must be chosen unless the default global origin is actually the rotation point of interest. Typically, the twist about the centerline-axis of the part is of interest. If $\vec{C}_*$ is a point on the centerline-axis of a part, then the geometric twist between two point clouds can still be computed as in (70). The shift between the data points varies with a change in the rotation point. Given a rotation point, $\vec{C}_*$, a shift between the data sets post-rotation can be calculated using equations (71) - (74), assuming the rotation angle has already been calculated. The key to this calculation is that the fiducial points that form $\vec{V}_1'$ must first be translated to the rotation point of interest, then rotated about the arbitrary point, and translated back. Once the points have been transformed in this manner, the shift between the two data sets can be calculated.

$$\vec{V}_2 = \vec{P}_1' - \vec{C}_* \tag{71}$$

$$\vec{V}_2' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{V}_2 \tag{72}$$

$$\vec{V}_3 = \vec{V}_2' + \vec{C}_* \tag{73}$$

$$\vec{V}_{shift} = \vec{P}_1 - \vec{V}_3 \tag{74}$$

Using the above equations, both form error and geometric error can be quantified in a MEMS part. These errors are analytically computed after least-squares registration is performed on both point clouds. Alternatively, the above equations can be used without first performing a least-squares registration. After calculating the rotation angle and measured shift between the two data sets using equations (68) - (74), these transform parameters can be applied to the second point cloud, resulting in a four-point registration scheme. The four-point registration scheme is a closed-form calculation that aligns the two sets of fiducial marks from each point cloud. This type of localization generates a graphical picture of what the actual point cloud would look like if the two scans were obtained in the same fixture setup (i.e. both scans taken in one inspection setup on the metrology hardware). Figure 22 illustrates a scene in which four-point registration has been applied to the top point cloud after the bottom point cloud has been least-squares registered to the bottom face of the part. It can be seen that the actual part has both a shift and rotation component with respect to the bottom point cloud about the axis of extrusion of the part. This type of analysis does not calculate the form error in the part; however, it provides an overall geometric picture of the actual shape of the MEMS part.

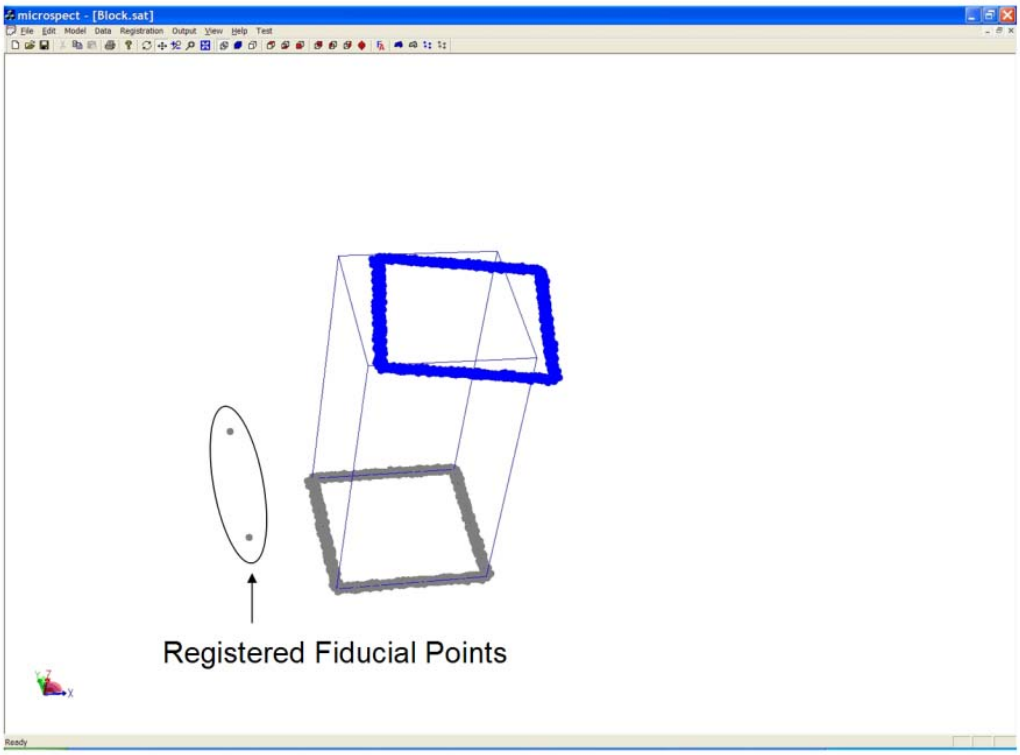

Figure 22: Four-point registration

The pseudo-code for the fiducial analysis procedure is shown in Table 8. The algorithm assumes that least-squares registration has already been applied to both the top and bottom point cloud of the analysis.

Table 8 : Pseudo-code for 4-pt fiducial analysis

```
vec1 = fiducial_pt2 - fiducial_pt1 //vector for bottom point cloud
vec2 = fiducial_pt4 - fiducial_pt3 //vector for top point cloud
Twist = angle_between_vectors(vec1, vec2)

//Calculate the shift given a rotation_pt (typically on the centerline
                                          axis of the CAD model)
temp_pt = rotate_point(fiducial_pt3, Twist, rotation_pt)
Xshift = temp_pt.X - fiducial_pt1.X
Yshift = temp_pt.Y - fiducial_pt1.Y
```

Geometric Parameter Fitting

Geometric parameter fitting is a useful method to analyze point clouds when a CAD model is unavailable. Additional parameters are added to the objective function before minimizing the errors. The framework previously established is, for the most part, untouched. The only additional parameters that need to be added are the geometric properties of the geometry being fitted. For example, for the least-squares registration of a sphere to an ideal CAD model, the parameters that are modified to find the best-fit location are translations in x, y, and z. If, instead, a point cloud is to be best-fit to a sphere with no CAD information available, one additional parameter must also be passed to the minimizer, the radius of the best-fit sphere. After minimization, the algorithm provides not only the best-fit translation (i.e. the center of the sphere) but also the best-fit radius for the point cloud. For this research, only analytical geometries are fitted. Spline-fitting is beyond the scope of this work and does not particularly apply to the application at hand. For the analytic geometries derived in this work, the additional parameters that need to be passed to the minimizer to perform a geometric best-fit are listed in Table 9.

Table 9: List of additional parameters for geometric fitting

Geometry	Additional degrees of freedom passed to minimizer
Line	None
Circle	Radius
Plane	None
Sphere	Radius
Cylinder	Radius
Cone	Radius and Half-included Angle
Torus	Major and Minor Radii

The deviation calculations remain the same; the only modification is that the minimizer is allowed to vary the geometric definition to which the point cloud is being fitted. Within the current framework, the data flow process for geometric parameter fitting is only slightly modified to include the additional geometric parameters that are passed to the minimizer as shown in Figure 23. Geometric parameter fitting is useful for fiducial registrations as is demonstrated in Chapter 5.

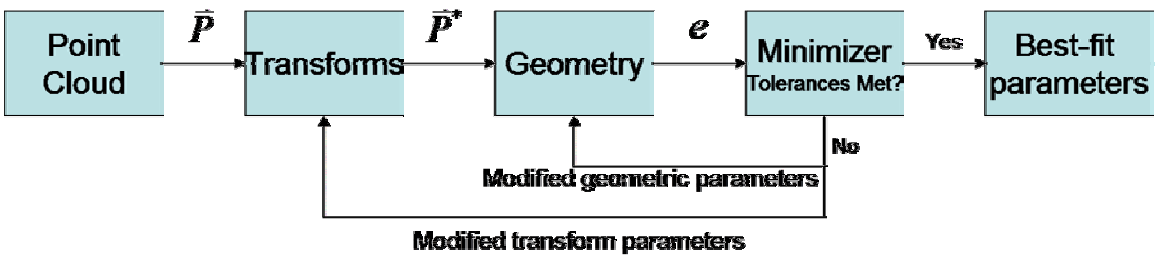

Figure 23: Modified data flow for geometric parameter fitting

Results Visualization

Once a point cloud has been registered the errors can be formatted for output by various means. Once the registration is completed simple statistics are outputted to the user such as the mean and standard deviation of the errors. For further statistical analysis, the deviations of the point cloud to the CAD model can be exported into other software for further processing. This research takes the approach of outputting a formatted text file with a list of the deviations for each point.

This research also implements algorithms to visualize the computed errors from the registration process. In traditional CAI programs, color maps are frequently used. This technique applies a gradient coloring scheme to each of the surfaces of the CAD model. In order to be more conducive to the 2-D/3-D framework developed in this work, however, whisker plots are generated to visualize the errors in the data. Figure 24 illustrates the concept.

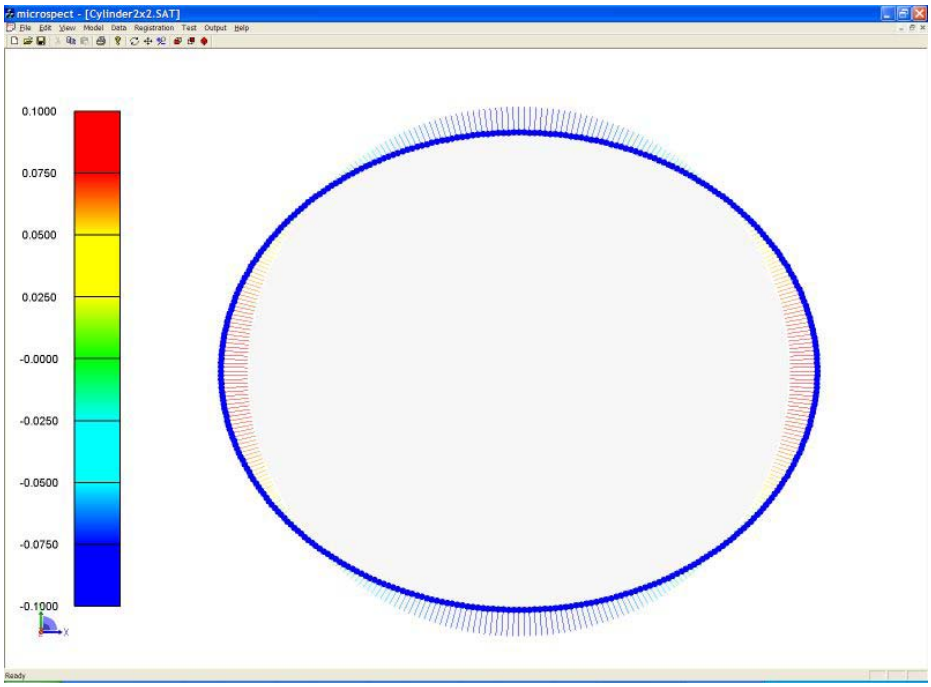

Figure 24: Whisker plot example

The first step in generating a whisker plot is to obtain the deviation vector from the minimizer. Using this deviation vector, a gradient scale can be generated from the smallest deviation to the largest deviation. Each deviation is then assigned an interpolated value within the gradient range. The most difficult step is then to determine the closest point on the entity in order to draw a line segment from the measured point to the entity. These calculations are extensions of the deviation calculations previously derived. The following sections detail the additional steps to determine the closest point on the entity to the measured data point.

Straight Line

In order to calculate the closest position on a straight line to a measured point, the line definition must be formulated as in Equation (26) for the trimmed geometry case.

The closest point is then calculated as in Equations (26) - (28). These equations are re-referenced in (75) - (77).

$$\vec{L}(u) = \vec{R} + \vec{D}u \tag{75}$$

$$u_0 = \frac{(\vec{P} - \vec{R}) \bullet \vec{D}}{\vec{D} \bullet \vec{D}} \tag{76}$$

$$\vec{P}_{closest} = \vec{R} + \vec{D}u_0 \tag{77}$$

It should also be noted here that the deviation calculation for a straight line does not distinguish between points that are "above" and "below" the line. In order to properly compute the color value of each deviation an arbitrary normal direction must be supplied to the whisker plot algorithm. If the normal direction is not the intended direction, this normal can be easily reversed to correct the color values of the whisker plot.

Circular Curve

For a circular curve, the closest position on the circle to the measured point is calculated using the definition and deviation calculation in Equation (6) as follows:

$$\vec{V}_1 = \vec{P} - \vec{C} \tag{78}$$

$$\vec{P}_{closest} = \vec{P} - \frac{\vec{V}_1}{\left|\vec{V}_1\right|} e \tag{79}$$

General Parametric Curve

Since the geometric modeling kernel is used to determine the deviation calculations for general parametric curves, finding the closest position on a parametric curve to a given point is also handled by the modeler. There is no direct method for calculating the closest point on an arbitrary parametric curve, and it is assumed that some type of variant of Newton's method is used to iteratively find the closest point.

Plane

Using the previous definition of a plane, the closest point on a plane is calculated with Equation (80).

$$\vec{P}_{closest} = \vec{P} - e * \hat{n} \tag{80}$$

Sphere

Calculating the closest point on a sphere is simply an extension of the 2-D circular curve case. The equations are identical and are shown in (81) - (82).

$$\vec{V}_1 = \vec{P} - \vec{C} \tag{81}$$

$$\vec{P}_{closest} = \vec{P} - \frac{\vec{V}_1}{\left|\vec{V}_1\right|} e \tag{82}$$

Cylinder

For a cylinder, one other vector must be defined in addition to the deviation calculation equations in order to calculate the closest position to a measured point. Namely,

$$\vec{V}_3 = \vec{P}_1 - \vec{P}_3 \tag{83}$$

Then, the closest point can be found using

$$\vec{P}_{closest} = \vec{P} - \frac{\vec{V}_3}{\left|\vec{V}_3\right|} e \tag{84}$$

Cone

The current cone deviation calculation computed in Equations (12) - (17) contains all of the necessary information for the closest point calculation. The closest position on the cone to the measured point is found with Equation (85).

$$\vec{P}_{closest} = \vec{P} - \frac{\vec{V}_n}{\left|\vec{V}_n\right|} e \tag{85}$$

Torus

For a torus, the closest point is calculated using the deviation calculation in Equations (18) - (21). Equation (86) shows the computation for determining the closest point.

$$\vec{P}_{closest} = \vec{P} - \frac{\vec{V}_3}{\left|\vec{V}_3\right|} e \tag{86}$$

General Parametric Surface

Again, the modeling kernel implements all inquiries related to the geometry of general parametric surfaces. The closest point on a given surface is calculated using the application programming interface (API) of the modeling kernel. Due to the closed

source nature of the product, no claims can be made as to the specifics of the closest point algorithm.

Whisker plot display

Once the closest position on an entity to a corresponding measured point is calculated, a line segment can be drawn from the measured point to the entity which gives a visual indication of the size of the errors. If the deviations are all relatively small, this line segment can be scaled to provide a more functional visual display. Alternatively, the framework allows the option of color each individual data point with its corresponding interpolated color instead of displaying the whisker plot.

Summary

The previous sections detail the necessary tools for the implementation of the methodology developed in this work. The fiducial analysis derived in this work serves as the critical step that provides new information for geometric characterization of high aspect ratio parts. The specific requirement for this analysis is the inclusion of fiducial information with each independent point cloud. This fiducial information should be identical between each point cloud. The geometry of the fiducial information can be any feature that has a unique point property. The unique point property can be any range of properties of a fiducial such as a center of a circle, center of a sphere, centroid of an arbitrary shape, etc. The results of the fiducial analysis will only be as good as the measured fiducials, so the fiducials should be known to have features that can be repeatedly measured with high confidence. The geometric parameter fitting algorithms

implemented in this work serve as ideal tools for calculating this point property of fiducial structures.

The developed methodology in this work can be used in many different ways. A typical process flow of the methodology is shown in Figure 25. First, the required data are loaded into the software platform (i.e. top and bottom point clouds and CAD model). Transforms are applied to the point clouds which allow for the manipulation of the point clouds for registration. Least-squares registration is then applied to the point clouds, independently. The results of the least-squares registration provides the form errors measured in the part. Geometric parameter fitting of fiducial information is typically the next step. The result will be the required points needed for the fiducial analysis. The fiducial analysis is then applied, which provides information about the 3-D geometric errors in the part. This process serves as the main contribution of this work. The last step typically performed is results visualization or deviation output for further statistical processing.

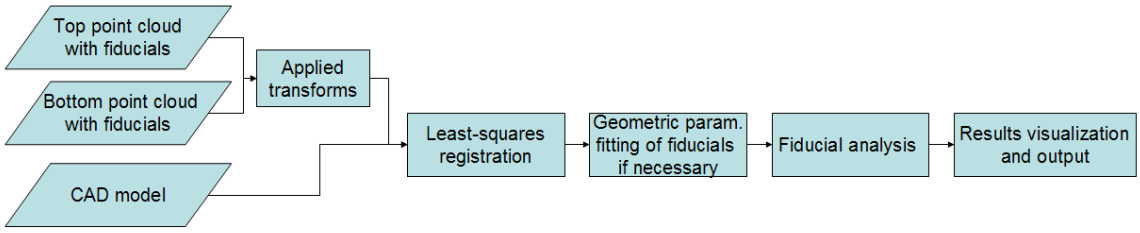

Figure 25: Typical flow of developed methodology

CHAPTER V

ANALYSIS AND VALIDATION

Overview

This chapter will present the analysis and validation for all of the algorithms and equations derived in this research. First, a method must be presented to analyze the developed algorithms. This method, ideally, will introduce no uncertainty of its own, allowing for the accurate characterization of the proposed algorithms. Once this method is established, testing of the deviation calculations is performed. Once the algorithms have been verified, testing of the algorithms with stochastic errors is performed to better simulate data obtained from actual metrology hardware. These tests also provide a measure of robustness for the algorithms. After the deviation calculations are verified, examples of point-to-entity assignments are given, highlighting possible errors with noisy data. The use of the algorithms derived for trimmed geometry is shown to eliminate some of these effects.

The next section verifies the fiducial analysis methods developed in this work. These algorithms are proven to be effective in determining geometric errors between multiple point clouds and are particularly useful for the analysis of high aspect ratio MEMS. The four-point registration algorithm is also shown to be useful in visualizing these geometric errors.

Finally, a series of case studies are presented to prove the applicability of the platform developed in this research. Test cases for CNC-machined parts, stereolithography parts, and a LIGA part are presented and analyzed.

Simulated Data

The foundation of the software platform developed in this research is the registration of a point cloud to a given geometric entity. In order to verify the developed algorithms, a method in which the best-fit of the data is known *a priori* must be used. Generating data from a CAD model is relatively straightforward and is the chosen method for this research. Data generated by this method can be calculated to the precision of the machine and is, thus, extremely accurate. Various levels of noise can be added to the data in order to better simulate data acquired from metrology hardware.

The first step in generating simulated data is to obtain the geometric definitions of each edge or face of a CAD model. Using the modeling kernel, the entity can be queried for the parameter range of the underlying geometry. In the ACIS modeling kernel, every face and edge can be parameterized. This parameterized interval can be subdivided, typically using even interval spacing, into the number of desired points for the simulated point cloud. These subintervals are then passed back to the geometric definition of the entity, and the corresponding position in three-dimensional space is returned by evaluating the geometry at the specified intervals. Figure 26 displays the parameterization lines of a spline surface using an equally spaced 10 x 10 grid. The

intersections of the parameterization lines represent the locations where the surface is evaluated for a position.

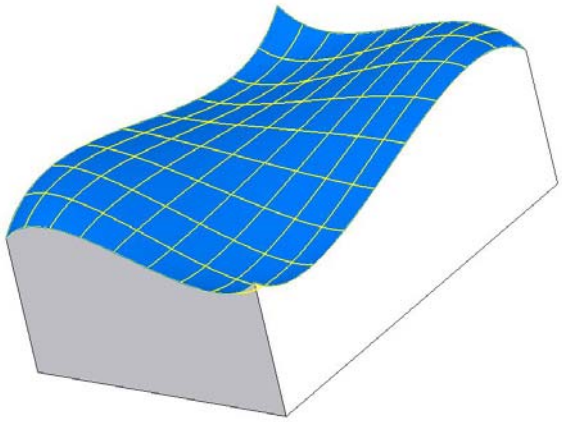

Figure 26: Using parameterized surface to generate simulated data

Simulated data which lie exactly on the given geometry are useful for testing the accuracy of the registration algorithms. For robustness testing, however, perfectly generated data give little insight into the performance of the algorithms when actual data from metrology hardware are used. Noise in the data is inherent to the data acquisition process. The simplest method to add noise to a data set is by adding a random value to each component of the point data (i.e. x, y, and z). The magnitude of this value is typically limited to some range, $n_{mag} \in [-x, x]$, resulting in a uniform distribution. However, data acquired from manufacturing processes typically fit curves represented by normal distribution curves (Kalpakjian 2003). The mean of the noise is still typically centered about zero, but a probability associated with the magnitude of the noise is used. The resulting distribution is then shaped in a traditional bell-shaped curve with a standard

deviation that represents the dispersion of the data. Gaussian noise is the type of noise used in this research.

Applying normally generated values of noise to a point cloud is necessary, but not sufficient, in order to impose a normal distribution to the calculated deviations of the geometry. The other critical factor is the direction of noise to the data. Since all the deviations are calculated with respect to the normal of the geometry, the noise must also be applied in this direction to result in a normally distributed error plot. Figure 27 illustrates an example of a surface with the normal directions used to apply the noise magnitude.

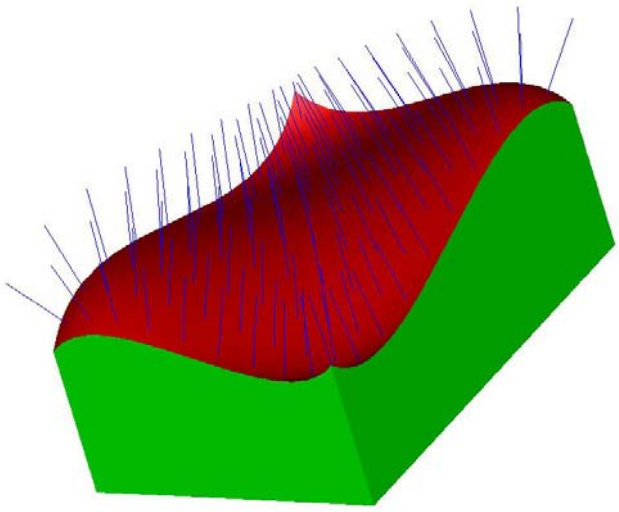

Figure 27: Normal directions used to generate noise

Another factor which impacts the outcome of registering data is the initial starting point of the point cloud. In typical CAI programs, the user manually positions the point cloud to a visually close fit. The initial guess for the transformation parameters are then

calculated from this manually-aligned position and passed to the minimizer. Different starting positions of the data will also be used to demonstrate the capabilities of the implemented registration routines.

Deviation verification

For each geometry type analyzed in this research, simulated data without noise is used to verify that the correct parameters of registration are calculated. Once the algorithms are proven to register perfectly generated data, noise is introduced in order to more accurately represent data acquired by physical metrology hardware. The improper assignment of noisy data is also discussed in the section below. In this chapter, the verification of straight line geometry deviation calculations is presented. The results for the other geometry deviation calculations are presented in Appendix A.

Straight Line

In order to verify the deviation calculations for straight line geometry, a CAD model is made which represents a face of a rectangular solid. This face contains four straight edges which will be used for the verification. Four thousand points are generated, which lie exactly on the lines (to the floating point precision of the machine). The registration routine developed in this work should find the best fit transforms and output deviations equal to zero for each point. The transform parameters that apply to this scenario are x-translation, y-translation, and z-rotation. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithms developed. Typically, the user will manually position the point cloud to a qualitative

good fit. The first of the five starting positions will represent an initial starting point that one can expect to be used in practice. The rest of the starting positions are relatively poor starting positions for registration, but demonstrate the effectiveness and robustness of the algorithms. Table 10 lists the starting positions of each point cloud before registration. Figure 28 graphically shows each transformed point cloud and the final registered location.

Table 10: Initial starting positions before registration for linear geometry

	X translation (linear)	Y translation (linear)	Z rotation (radians)
Optimal Fit	**0.0**	**0.0**	**0.0**
Initial Guess A	0.01	0.01	0.01
Initial Guess B	0.5	0.35	0.1
Initial Guess C	0.4	-0.25	-0.2
Initial Guess D	-0.35	0.45	0.3
Initial Guess E	-0.25	-0.5	0.35

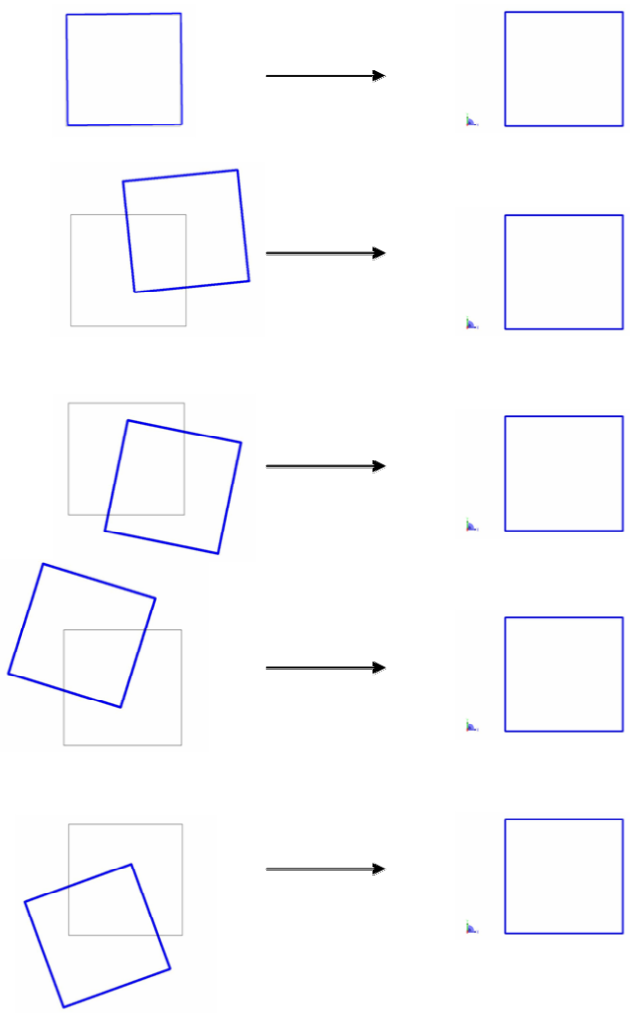

Figure 28: Initial guesses and registered point clouds for linear geometry

As expected, the point clouds are coincident with the nominal CAD geometry after registration. The summary information for each case is listed in Table 11.

Table 11: Deviation summary for test cases for linear geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	0.000000	0.000000	-0.000000	0.000000
Initial Guess B	-0.000000	0.000000	-0.000000	0.000000
Initial Guess C	-0.000000	0.000000	-0.000000	0.000000
Initial Guess D	-0.000000	0.000000	-0.000000	0.000000
Initial Guess E	-0.000000	0.000000	-0.000000	0.000000

With perfectly simulated data, the deviations are all reported to be zero, up to six reported decimal places. The CAD model is on the order of 1 unit; therefore, the errors associated with the part are reported to the millionths of a unit. The actual computations, however, are used with double precision numbers (15 digits of precision), and the residuals from the minimizer are actually zero up to the thirteenth decimal place, which for all practical purposes is zero. The only other issue to be noted about these results is that the mean is reported as negative or positive depending on the starting position. This observation is due to the direction associated with the deviation calculation and the tolerance conditions of the minimization algorithm.

In order to better simulate real world conditions, the registration procedure is validated with simulated noise as discussed in the previous section. The first set of data is generated with a standard deviation of 0.001 units. If these units are taken to be mm, the amount of noise in the generated data set is on the same order as the repeatability of most coordinate measuring machines, which are considered to be very accurate data acquisition hardware. The same procedure for analysis is used as for the perfectly generated data above. Table 12 summarizes the results.

Table 12: Deviation summary for 0.001 unit noise for linear geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made	-0.000009	0.000986	-0.003407	0.003353
Initial Guess A	-0.000010	0.000984	-0.003431	0.003381
Initial Guess B	-0.000010	0.000984	-0.003431	0.003381
Initial Guess C	-0.000010	0.000984	-0.003431	0.003381
Initial Guess D	-0.000010	0.000984	-0.003431	0.003381
Initial Guess E	-0.000010	0.000984	-0.003431	0.003381

From this analysis, the minimizer is shown to be repeatable with respect to an induced noise factor. All initial starting points result in the same fitting solution. The numbers agree with the as-made deviations, with the standard deviation agreeing to within two millionths of a unit. The deviation vector is the same for each registered case up to at least the tenth decimal place. Figure 29 shows the histogram and QQ-plot for the as-made data set and one of the registered data sets. From this figure, the registered data set is shown to have a nearly identical standard deviation and normal distribution fit as the original as-made data.

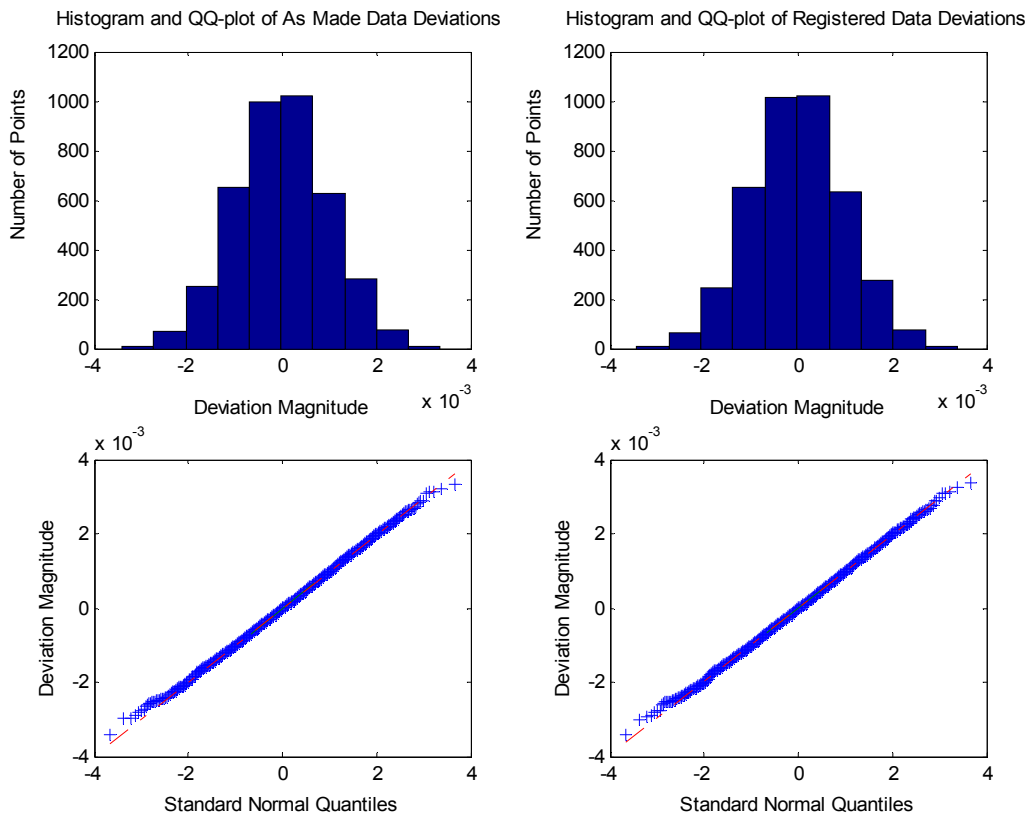

Figure 29: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for linear geometry

Increasing the noise level to a standard deviation of 0.01 units (i.e. an order of magnitude larger than the last set) produces a data set that is significantly noisier and can visually be seen as shown in Figure 30. Noise of this magnitude can sometimes be seen in data from edge detection techniques. Artifacts and outliers are common in edge detection techniques, and thus the analysis of the linear fitting algorithms is tested with this simulated case. Table 13 summarizes the output of the registration procedure.

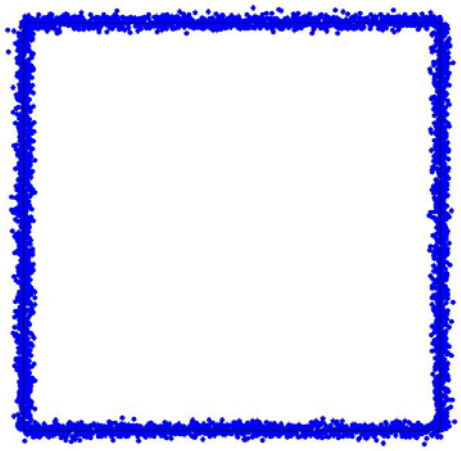

Figure 30: Data set with 0.01 units of noise for linear geometry

Table 13: Deviation summary for 0.01 unit noise

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000199	0.010038	-0.030575	0.039463
Initial Guess A	0.000006	0.009959	-0.030536	0.038106
Initial Guess B	0.000006	0.009959	-0.030536	0.038106
Initial Guess C	0.000006	0.009959	-0.030536	0.038106
Initial Guess D	0.000006	0.009959	-0.030536	0.038106
Initial Guess E	0.000006	0.009959	-0.030536	0.038106

Here again, the registration process is repeatable when the initial guess for the minimization routine is varied. The standard deviations of the as-made data and the registered data sets are within seven millionths of each other. The maximum and minimum errors for the unregistered set and registered sets are also comparable. The mean of the errors for the as-made and registered cases differ by two orders of magnitude. This result will be discussed at the end of this section. Figure 31 shows that

both the unregistered and registered residual errors in the data both fit well to a normal distribution, as they should.

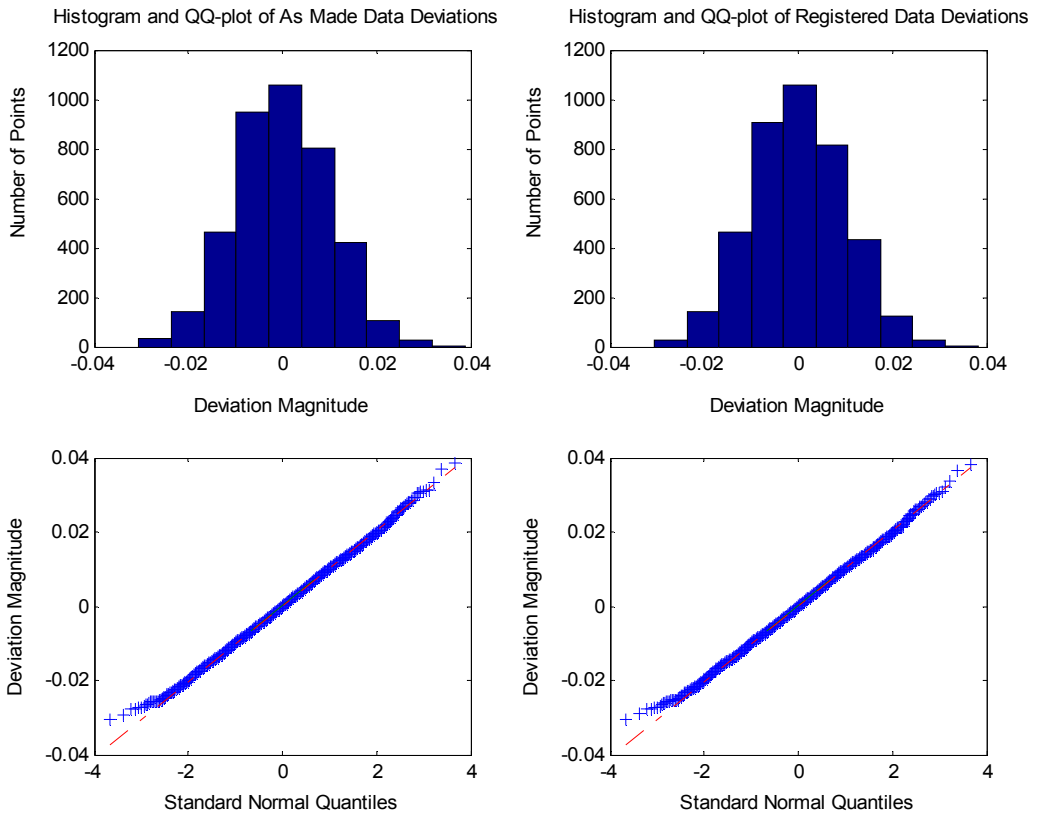

Figure 31: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for linear geometry

With the given noise induced on the data set, an interesting result occurs during the point assignment step of the registration routine. Approximately 40 of the 4000 points are incorrectly assigned to the underlying CAD geometry, regardless of the point assignment technique (i.e. exhaustive or sweep method). Figure 32 shows a partial whisker plot of the deviations calculated after registration.

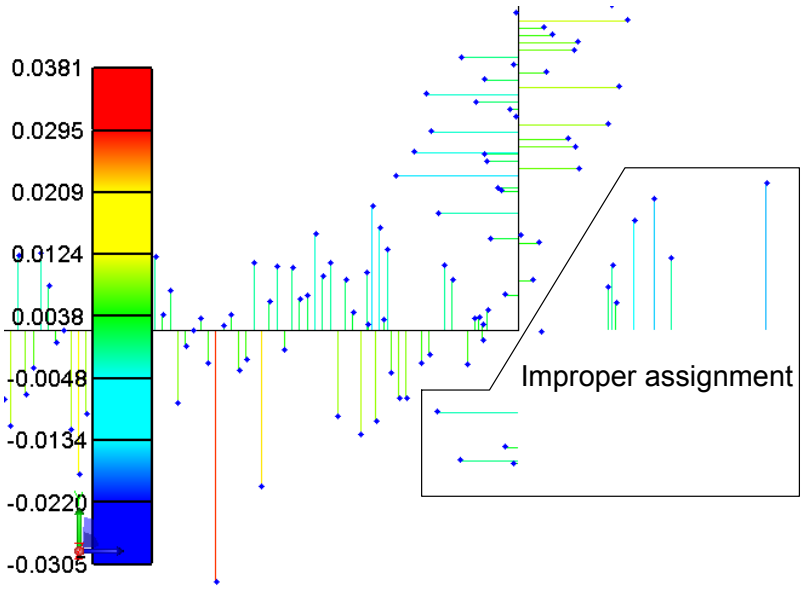

Figure 32: Improper assignment of points to underlying CAD geometry

At the corners of the CAD model, some of the points are assigned to line geometry that is actually not present. The reason for this error is because of the definition of the line used for the deviation calculation. As discussed in Chapter IV, the line definition assumes an infinite line in space. If data are sufficiently noisy, it is possible that the minimum deviation is calculated to part of a line definition that is not part of the CAD edge. In order to account for this type of error, the closest point on the line must be calculated, including the end points. The derivations for trimmed geometry deviations were also derived in Chapter IV. There can be a significant computational penalty added if trimmed geometry is used instead of the typical definition of a line; however, if the data are sufficiently noisy, incorporating trimmed geometry may be the most accurate method for registration. Figure 33 displays a partial whisker plot showing how the points are correctly assigned to the trimmed geometry definitions when using the deviation derivations for trimmed linear geometry.

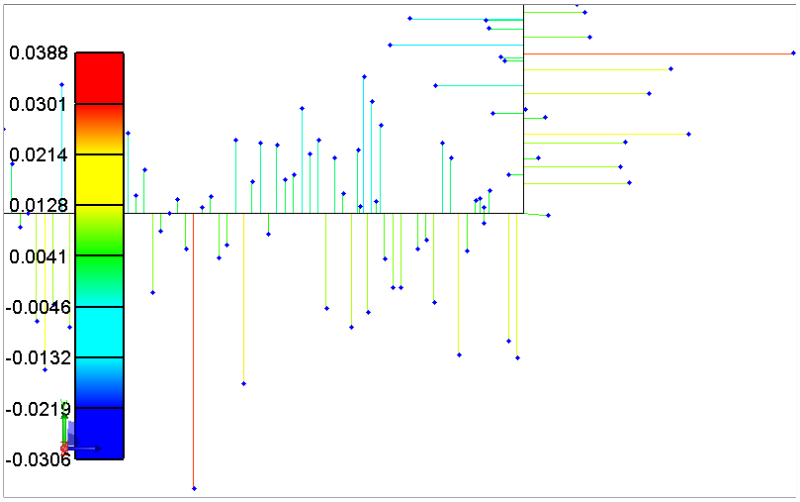

Figure 33: Proper assignment of points to underlying trimmed geometry

The effect of improperly assigned points is heavily dependent on the density of data in the point cloud, the standard deviation of the errors (i.e. how noisy the data are), and the number and types of entities in the CAD model. Improperly assigned points will only arise at intersections of entities (e.g., edge intersections and face intersections), and depending on the noise level in the data and type of geometry, improperly calculated deviations may or may not occur. For the particular case at hand, Table 14 summarizes the differences between using trimmed geometry for the two different levels of noise investigated.

Table 14: Trimmed vs. untrimmed calculations for linear geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made 0.001 unit noise	-0.000009	0.000986	-0.003407	0.003353
Initial Guess A Untrimmed	-0.000010	0.000984	-0.003431	0.003381
Initial Guess A Trimmed	-0.000009	0.000985	-0.003432	0.003382
As-made 0.01 unit noise	0.000199	0.010038	-0.030575	0.039463
Initial Guess A Untrimmed	0.000006	0.009959	-0.030536	0.038106
Initial Guess A Trimmed	0.000200	0.010031	-0.030563	0.038752

The above table shows that for the 0.001 unit noise, the difference between the untrimmed and trimmed deviation calculations are negligible. As the noise level increases, the improperly assigned points have more of an influence on the statistics of the fit. The minimum and maximum errors are still comparable, but the mean of errors is computed to be better than expected. This observation is due to the fact that the improperly assigned points actually report smaller deviations than if they were assigned to the correct entity.

Four-Point Registration and Fiducial Analysis

The least-squares registration analyses up to this point can be applied to any point cloud and any CAD model (2-D or 3-D). Besides least-squares registration, the other major analysis tool developed in this work is a four-point fiducial analysis scheme. This

type of analysis is developed specifically for MEMS inspection in order to provide information that is currently not obtainable using existing analysis methods.

The four-point registration algorithm is used to visualize any gross geometric errors in MEMS devices. Having two separate point clouds representing the top and bottom scans of a MEMS part, four-point registration can be used to quickly visualize any shift or rotation along the axis of the part. The primary requirements for this type of registration are four known data points, two belonging to each point cloud as shown in Figure 34.

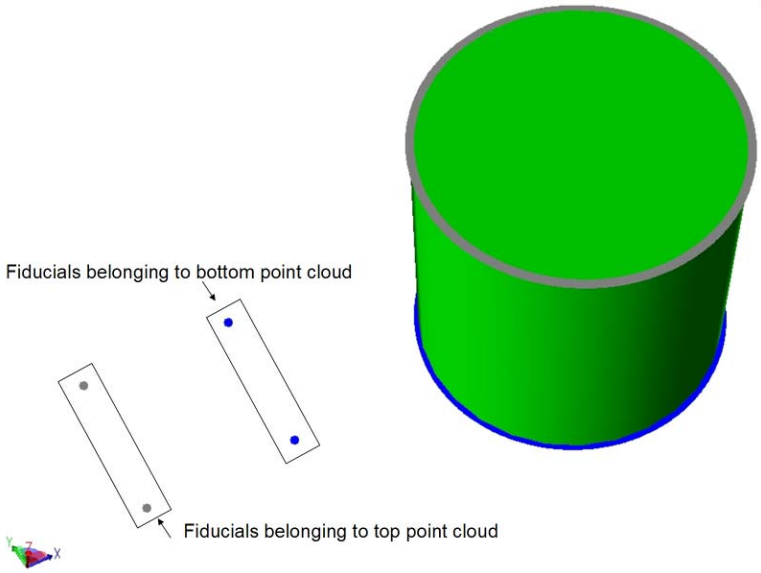

Figure 34: Illustration of fiducials with corresponding point clouds

In the above example, a CAD model of a cylinder of unit radius is shown with two simulated point clouds including fiducial data. Visually, the two point clouds appear to have relatively acceptable form error with respect to the circular geometry. A least-squares registration provides the actual quantitative numbers. However, assuming the

two sets of fiducial points in this scenario are one and the same, an obvious geometric error exists. The data in this case was generated to simulate a unit shift in the x-direction. Using the four-point registration algorithm developed in Equations (68) - (74), the top point cloud can be registered to the bottom point cloud as shown in Figure 35.

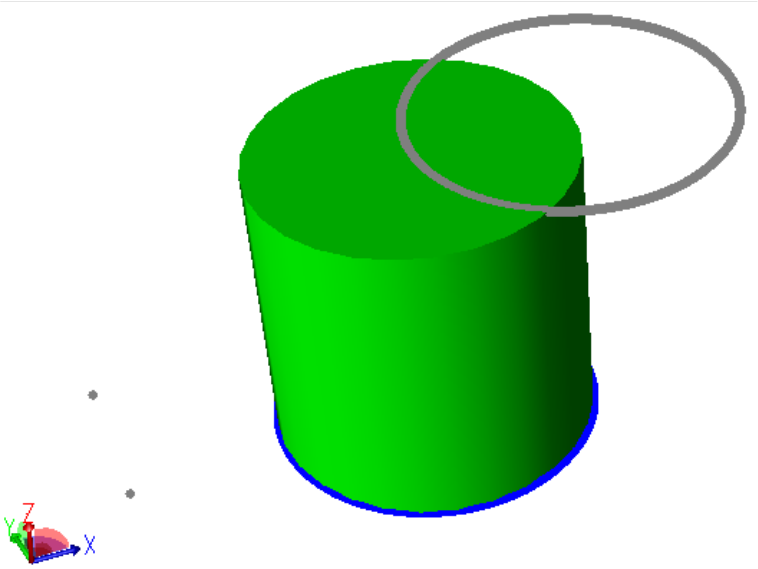

Figure 35: Four-point registration of top and bottom point clouds

Visually, a shift in the part can be seen after the four-point registration is applied. To better visualize the shift in the part, a model representing the physical part simulated in this analysis is shown in Figure 36.

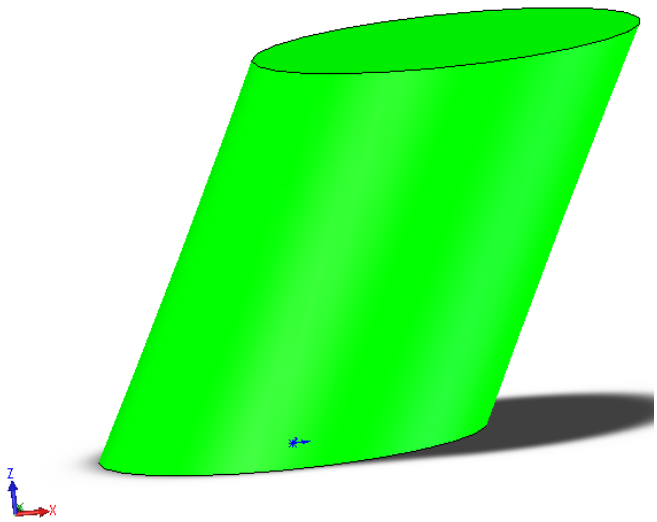

Figure 36: Model of part represented by the two sets of point clouds

For a more quantitative answer to the geometric error in the part, the shift can be determined using the same equations derived in (68) - (74), but without actually applying the transform parameters to the top point cloud. Instead, the information is presented to the user in quantitative form as shown in Figure 37.

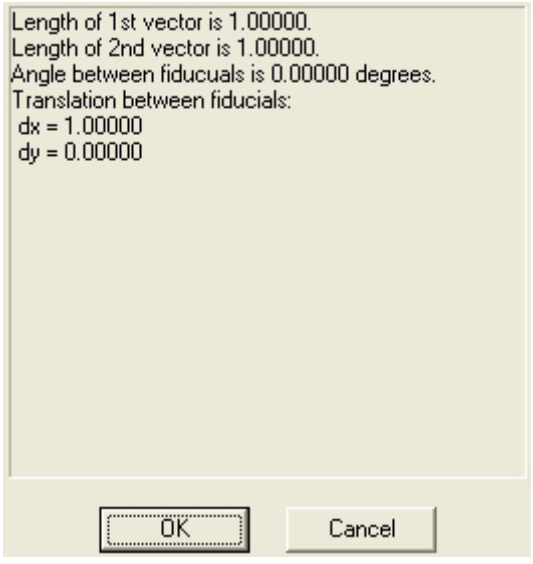

Figure 37: Quantitative output of fiducial analysis

For the current simulation, the results show exactly the intended effect, a shift in the x-direction of one unit. Twists in a MEMS part can also be detected using the derived fiducial analysis. A CAD model of a rectangular block and two point clouds representing scans of the top and bottom edges are modeled to verify the calculation of twist in a MEMS part. Figure 38 shows the generated CAD model and data sets.

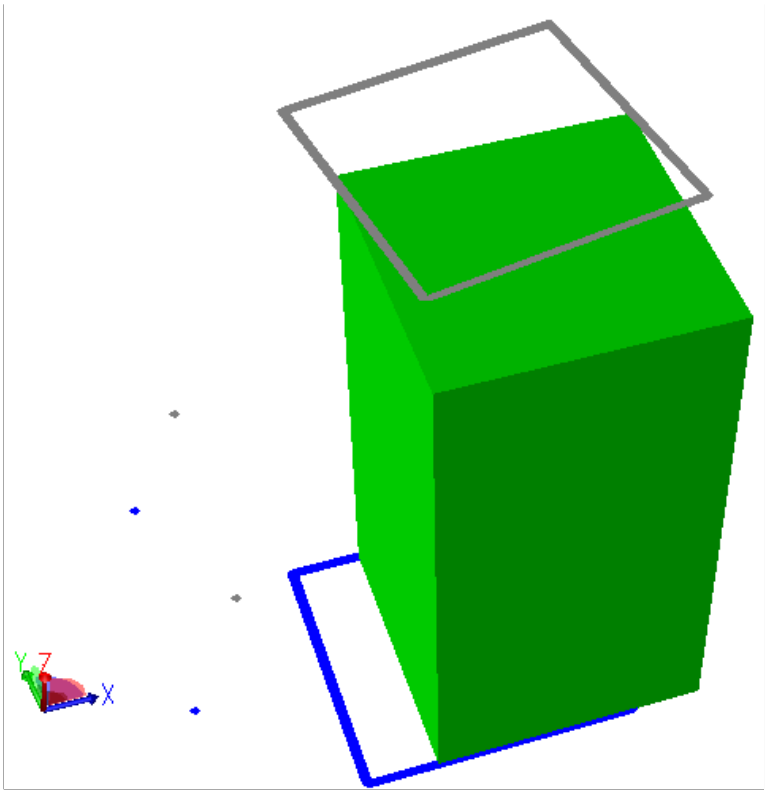

Figure 38: Illustration of simulated twist error

Using least-squares registration for both point clouds, the form error of the part is quantified. The resulting least-squares registration is shown in the following figure.

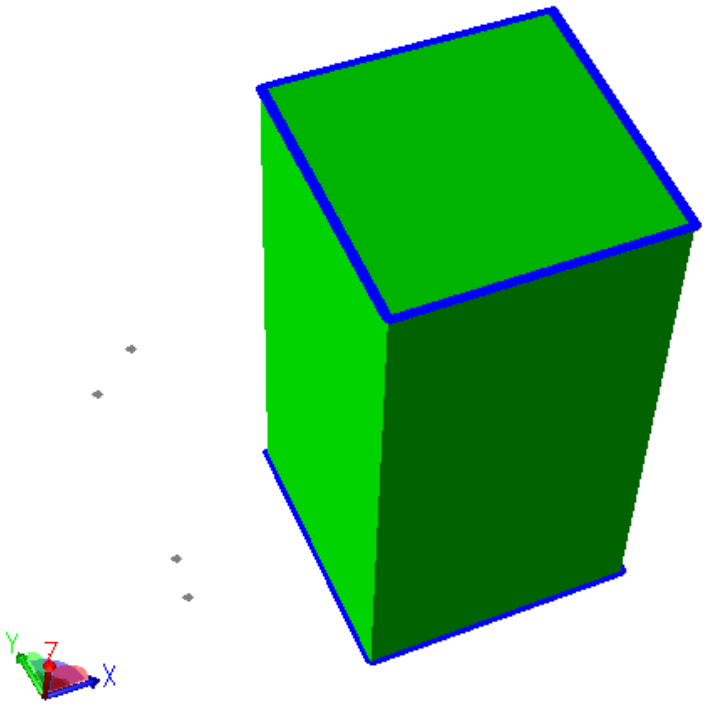

Figure 39: Least-squares registration for both point clouds

Since the data is generated to lie exactly on the edges of the part, the statistics from the least-squares registration all converge to zero. From the alignment of the fiducials, however, it is apparent that there is a twist in the part, since the vector directions of each fiducial set are not parallel. Using the four-point registration scheme, the twist can be visualized as shown in Figure 40.

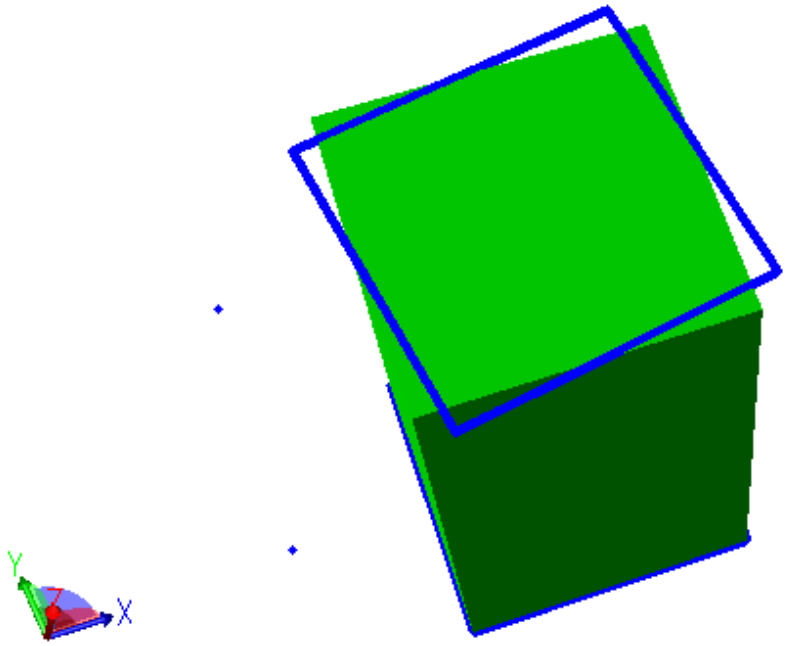

Figure 40: Visualization of twist error

From this visualization, the physical part that the data represents is shown to have only a twist about the z-axis. To give a better picture of what the point clouds actually represent, a model of the physical part that is simulated is created and shown in Figure 41.

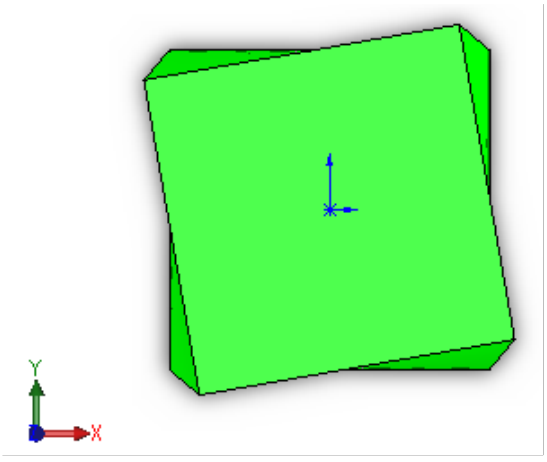

Figure 41: Model of part represented by the two sets of point clouds (top view)

For a more quantitative analysis, the fiducial analysis is done without applying the transform parameters and the results are reported to the user as shown in Figure 42. The results are concurrent with the as-designed model and data. There is an induced ten degree rotation between the two data sets with respect to the fiducial points.

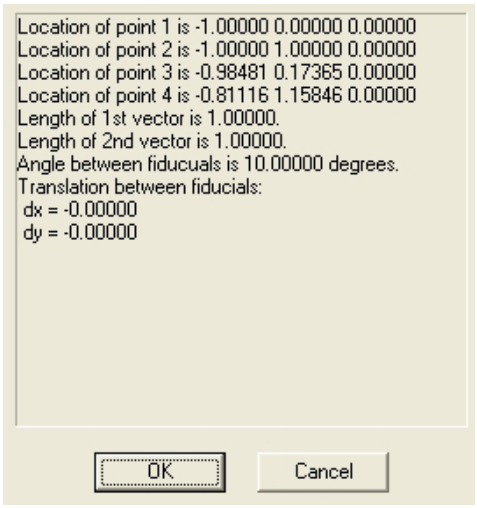

Figure 42: Quantitative output of fiducial analysis

The four-point fiducial analysis can also calculate information from point clouds that are both shifted and rotated with respect to each other. Care has to be taken, however, in selecting the rotation point about which to calculate the shift measurement. Due to the procedural nature of transformations, a rotation followed by a shift is not necessarily the same as a shift followed by a rotation. The previous block CAD model is used to analyze point clouds which contain both a shift and twist error. In this example, the top point cloud is shifted 0.35 units in the x-direction, 0.30 units in the y-direction, and rotated 15.0 degrees about the centerline of the part. The point clouds are simulated

with no noise and are least-squares registered to the part. After the least-squares registration the following figure is generated.

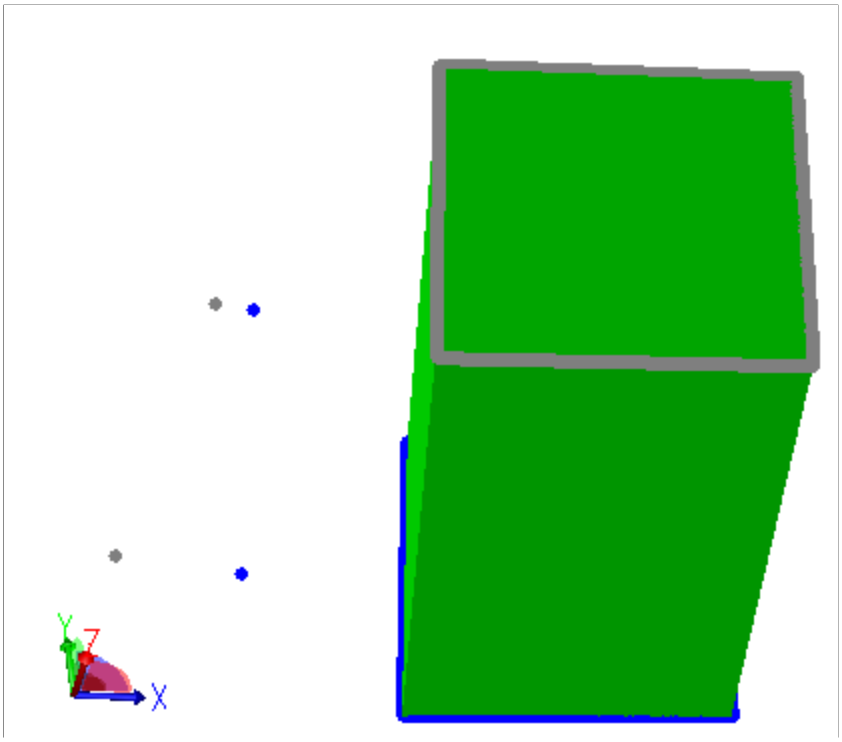

Figure 43: Point clouds with both shift and twist errors

The least-square registration produces, as expected, a zero deviation vector (i.e. perfect fit). The actual data fit well to the form of the edges of the part. There is a discrepancy in the fiducial points however, and the fiducial analysis algorithm reports the following results.

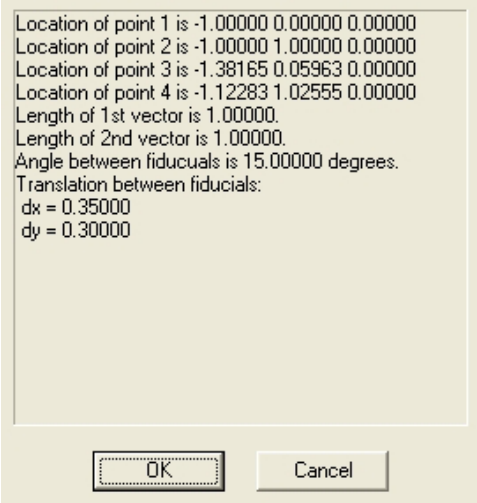

Figure 44: Results of fiducial analysis

From the fiducial analysis, the top point cloud is shown to be shifted in the x-direction by 0.35 units, y-direction by 0.30 units, and rotated by 15.0 degrees with respect to the bottom point cloud. These results concur exactly with the simulated errors. To provide a better picture of the actual errors in the simulated physical part, a CAD model which represents the point cloud data is shown below.

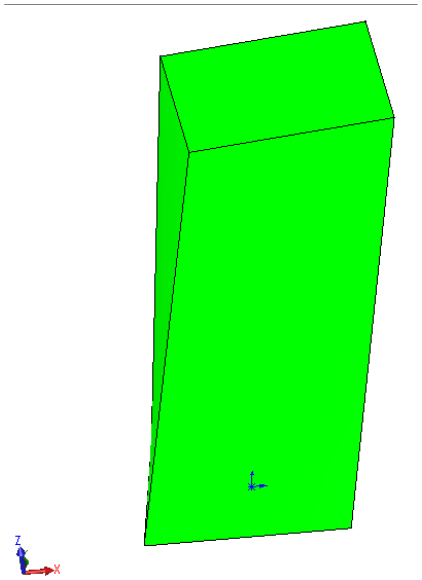

Figure 45: Model of part represented by the two sets of point clouds

The previous examples illustrate the usefulness of fiducial analysis. Using only least-squares registration for each point cloud may identify minimal errors. However, if fiducial information is available other geometric errors can be measured that previously were unrealized. The following case studies further augment this point.

Case Studies

The remainder of the chapter consists of case studies devised to utilize the developed software platform and illustrate the usefulness of the algorithms developed. The first case study consist of various test parts fabricated using both a CNC mill and an SLA system. The test parts are representative of macro-scale MEMS parts with induced errors of current interest. The last case study is of a LIGA device fabricated and measured at Sandia National Laboratories in Livermore, CA. This case study illustrates

the current solution to inspecting high aspect ratio parts using the developed software platform, and proposes a new technique for acquiring top and bottom scans of a part.

CNC and SLA Case Study

A series of parts were fabricated using a CNC mill and an SLA which contain errors that are of interest in MEMS fabrication. These manufacturing techniques are chosen because the processes allow for the easier incorporation of the desired errors than micro-fabrication techniques. Another reason to use parts of this scale is because of the metrology tools available to digitize the parts. Figure 46 and Figure 47 are photos taken of the final parts.

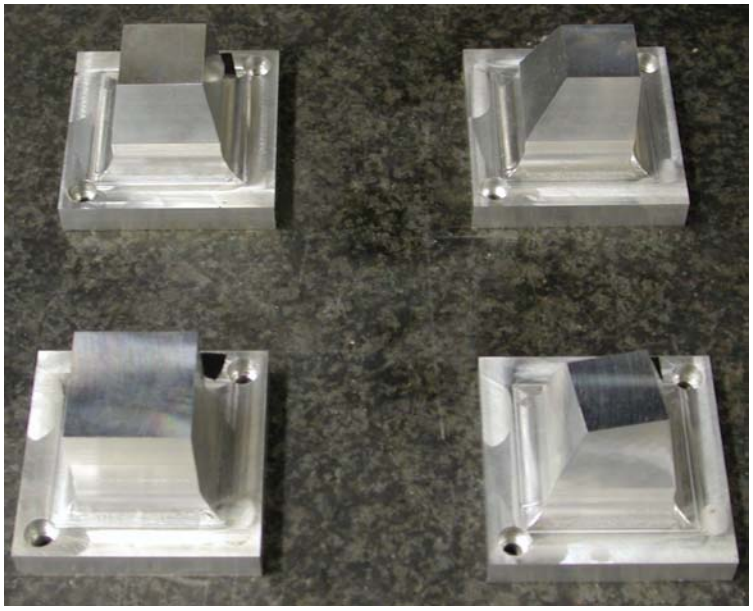

Figure 46: CNC milled test parts

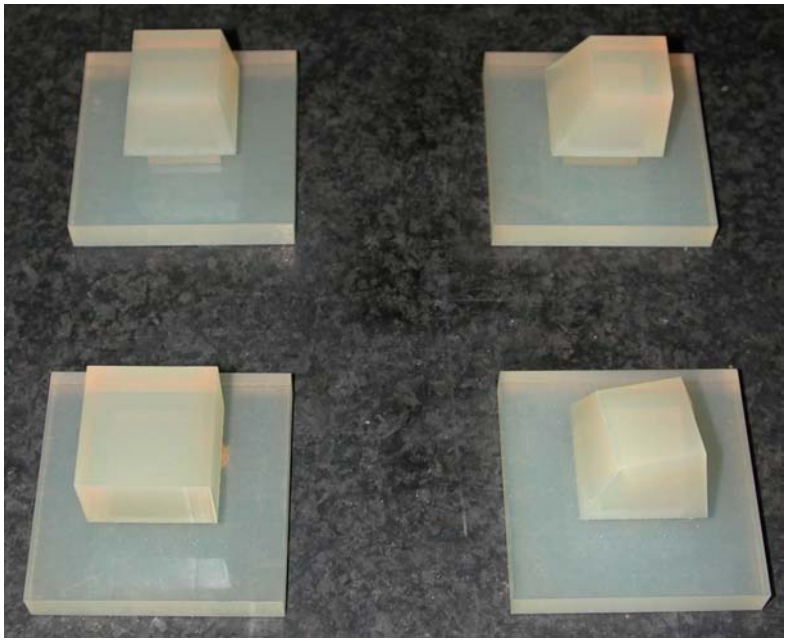

Figure 47: SLA fabricated test parts

The nominal geometry that is to be analyzed is an extruded square profile, which is shown in the lower left of the taken photographs. The other three parts have geometries representative of shift, scale, and twist errors from the ideal rectangular block. The nominal block has dimensions of 30 x 30 x 20 mm. Each part sits on top of a base which is used for alignment purposes on the CMM. In order to simulate edge data taken from a vision system, which was previously discussed to be the primary method for acquiring data from MEMS devices, various cross-sections of the part are digitized by the CMM. Digitizing of the cross-sections of the part is done by programming the CMM to collect data at various z-heights along the axis of extrusion of the part. Each cross-section will be a separate point cloud for analysis. Once the various cross-sections have been digitized, a corresponding CAD model is modeled in which the edges lie in the same plane as the acquired data. An example of a scene with the CAD model and two imported cross-sections is shown in Figure 48.

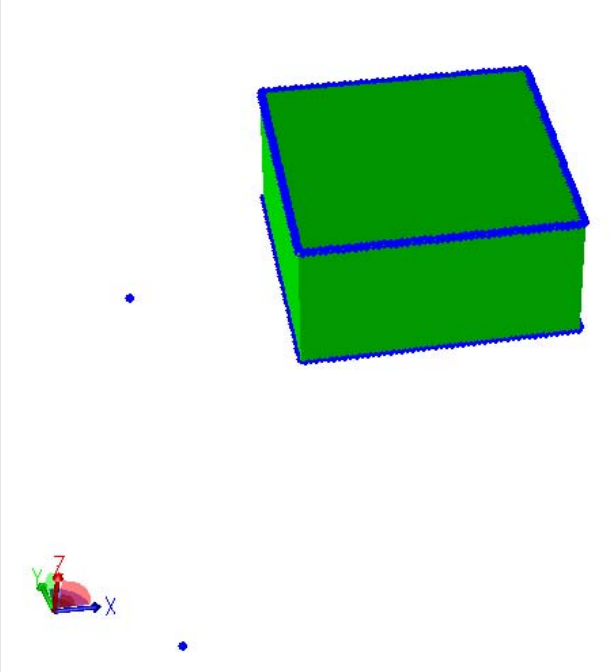

Figure 48: Nominal square geometry for case study with measured data

The above geometry represents the two cross-sections of the part that were inspected by the CMM. The two point clouds are the actual measured points of the two cross-sections of the part. The points not located close to the geometry represent the lower-left and upper-left corners of the reference base, which are used for fiducial information. The above scene is analogous to importing edge data inspected from a MEMS part along with the CAD model, with exception to the fiducial points. Generating fiducial points with a vision technique will be addressed in the final case study in this chapter.

As previously mentioned, a CNC mill and an SLA system were used to manufacture the same set of parts. For the data shown in Figure 48, the results of the least-squares registration of the cross-sectional analyses are shown in the following table. All units are in mm and reported to the resolution of the CMM (0.0001 mm).

Table 15: Results of least-squares registration for ideal part

	Mean of errors	Std. Deviation	Min. Error	Max. Error
CNC Top	-0.0504	0.0047	-0.0527	-0.0235
SLA Top	-0.1551	0.0229	-0.1864	-0.0430
CNC Bottom	-0.0319	0.0031	-0.0416	-0.0226
SLA Bottom	-0.1765	0.0213	-0.2037	-0.0531

From the results, it is clear that the machined case study has better tolerances than the SLA part. Clarification should be made about the min and max error numbers. The min error is calculated by finding the result with the absolute smallest error, irrespective of the sign of the error. Likewise, the max error is the deviation with the largest magnitude, irrespective of the sign of the error. From the reported errors, both test parts are undersized from the nominal geometry. For the machined part, this error can be caused by various factors such as tool diameter, starting cut location, etc. It should be noted, however, that the standard deviation is very small for both top and bottom analyses, meaning the part has good linear form. A simple tool offset correction should result in a part that is more accurate than the one currently being analyzed.

The SLA part has errors an order of magnitude larger than the machined case test part. Errors in SLA parts can be induced from many simple and complex factors, including hardware and software settings. Adjusting some of the build parameters of the parts may correct some of the measured errors.

Using the fiducial information for the parts, geometric errors in the part can be analyzed. With the current data sets, there should be small translation and rotation errors because the data are coming from a physical part representative of the nominal geometry.

Figure 49 shows the results of the fiducial analysis. The screenshot on the left is from the CNC part and the one on the right is from the SLA part. Again, the machined part is shown to have better tolerances with respect to the inspected cross-sections. Both show small twist and shift errors, as expected. All units are in mm and only the first four digits after the decimal are significant.

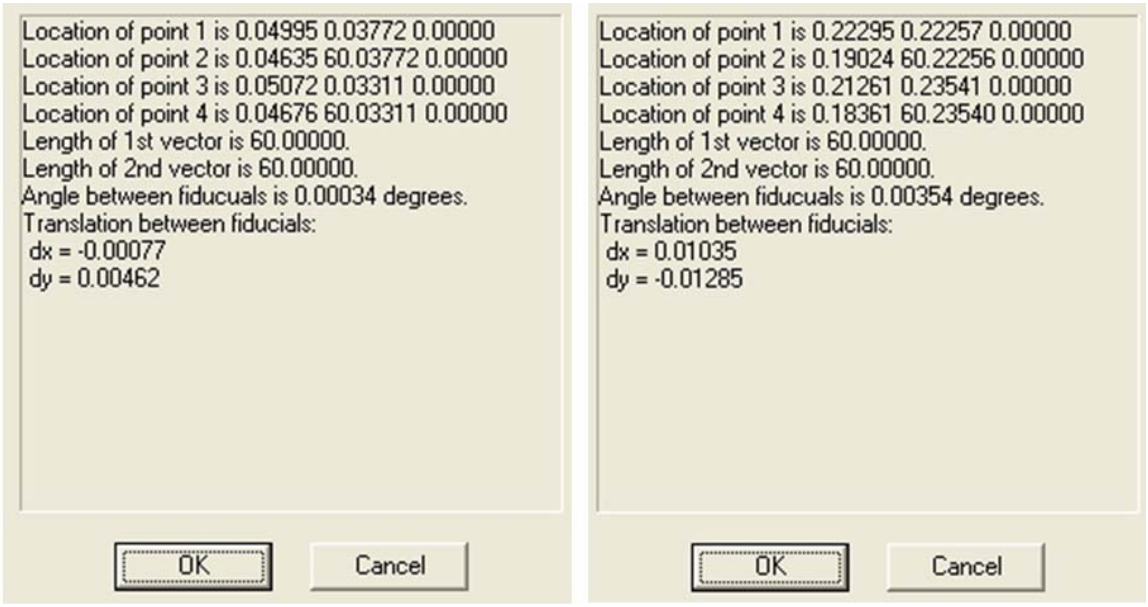

Figure 49: Results of fiducial analysis for ideal block

The second set of test parts analyzed has an induced draft along the z-axis of the part. These two parts are shown in the upper-left corners in Figure 46 and Figure 47. After importing the ideal geometry with the two point clouds, the following scene results:

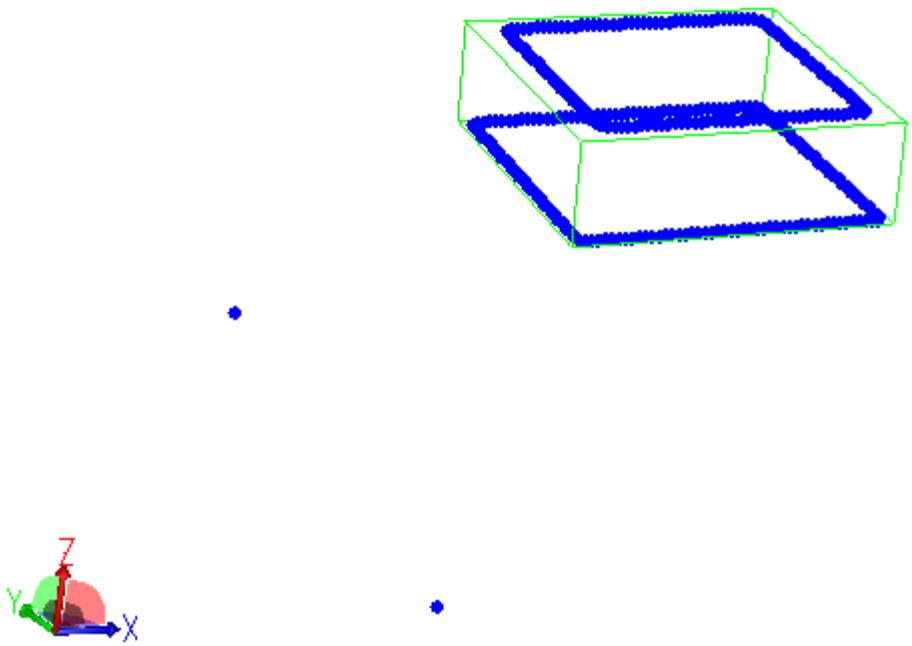

Figure 50: CAD model and point clouds with induced scale errors

In order to analyze the form errors in the produced part, least-squares registration is used on the two cross-sections of interest. From the whisker plots of each point cloud, it is obvious the measured part contains a taper error along the z-axis of the part. Figure 51 illustrates the whisker plot of the top point cloud which contains the greater scale error of the two point clouds.

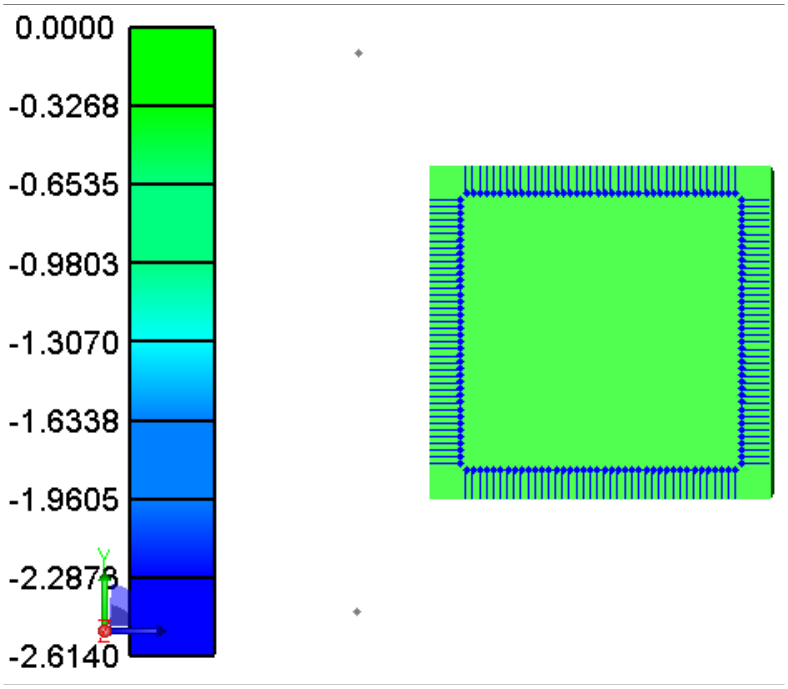

Figure 51: Whisker plot of top point cloud after localization for CNC milled part

The summary from the least-squares registration is shown in Table 16. The theoretical errors were calculated using the CAD model of the as-manufactured part compared to the ideal CAD model. All units are in mm, reported the accuracy of the CMM.

Table 16: Summary of results for drafted test part

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Theoretical Error - Top	-2.6250	0.0000	-2.6250	-2.6250
CNC Top	-2.6023	0.0033	-2.6140	-2.5928
SLA Top	-2.7831	0.0105	-2.7961	-2.7385
Theoretical Error - Bottom	-0.9000	0.0000	-0.9000	-0.9000
CNC Bottom	-0.8752	0.0028	-0.8886	-0.8686
SLA Bottom	-1.0669	0.0111	-1.0823	-1.0288

From the results, both parts show agreement with the theoretical error values. The CNC milled part has the tighter tolerances of the two parts, being within 0.0250 mm of the expected mean. The data taken from the part is also less noisy than the data from the SLA fabricated part, which can be seen from the lower standard deviation. The two parts should contain minimal errors due to shift or rotation between the two point clouds. Using the fiducial analysis techniques developed in this work, these errors can be quantified and are shown in Figure 52. The analysis for the CNC milled part is on the left, and the analysis for the SLA manufactured part is on the right. Both parts show little shift or rotation between the two point clouds. All units are in mm and the first four digits past the decimal are significant.

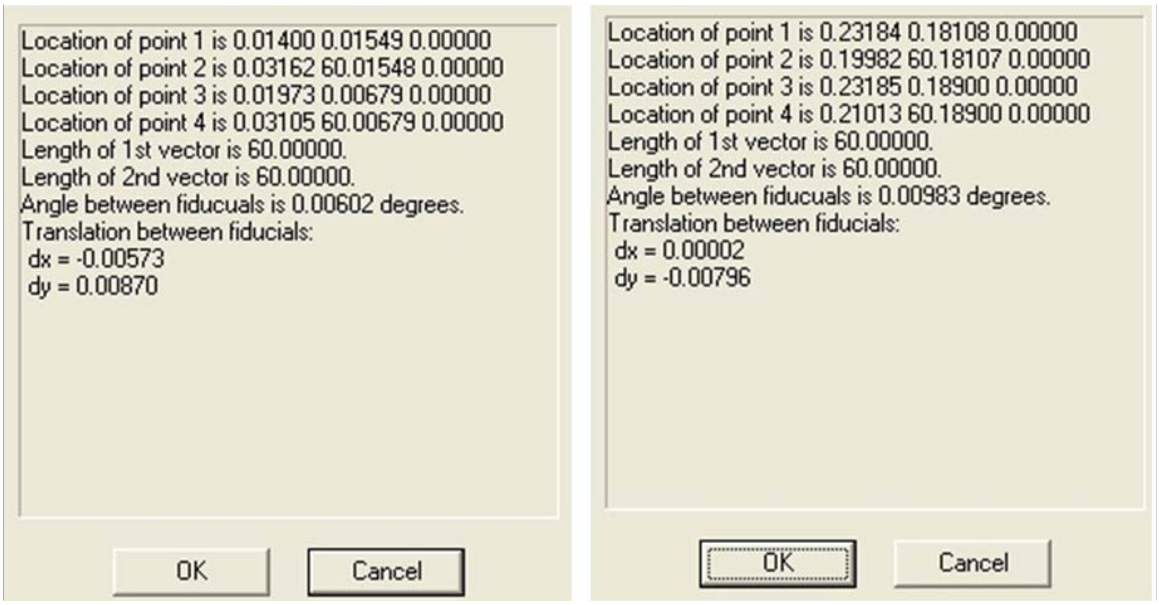

Figure 52: Results of fiducial analysis for drafted block

The next two test parts analyzed were manufactured to have a noticeable rotation error when compared to the nominal block geometry (shown in the lower right of Figure 46 and Figure 47). The CAI scene with the CAD model and imported data is shown in Figure 53.

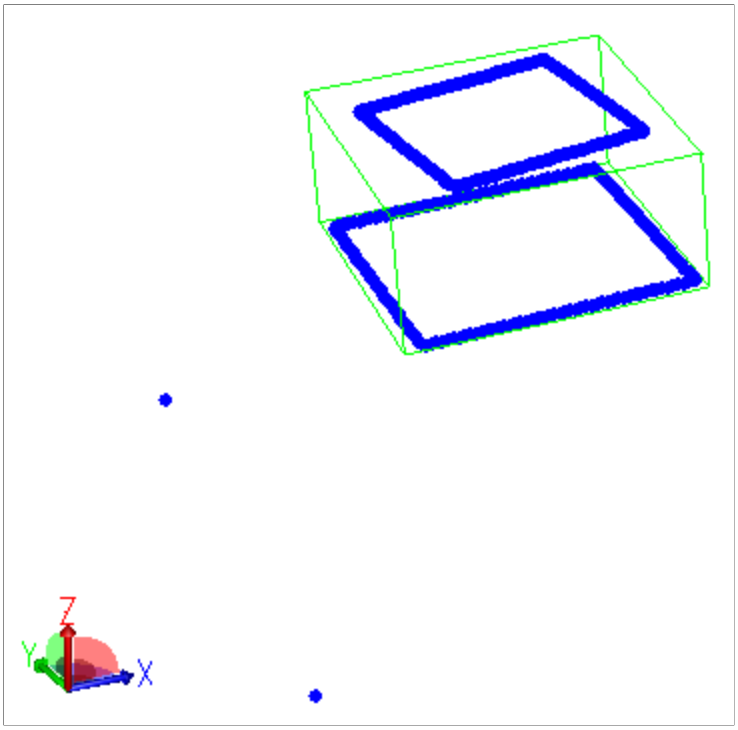

Figure 53: CAD model and point clouds with induced rotation error

The results for the least-squares registration for the point clouds are shown in Table 17. The theoretical errors are calculated using the geometry from the ideal CAD model and the error-induced CAD model. All units are in mm and are reported to the accuracy of the CMM.

Table 17: Summary of results for rotated test part

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Theoretical Error - Top	-4.9500	0.0000	-4.9500	-4.9500
CNC Top	-4.9803	0.0509	-4.9660	-4.9067
SLA Top	-5.0545	0.1142	-5.2201	-4.8468
Theoretical Error - Bottom	-1.3050	0.0000	-1.3050	-1.3050
CNC Bottom	-1.2898	0.00546	-1.3165	-1.2781
SLA Bottom	-1.4852	0.03485	-1.5308	-1.3360

From the results presented in the table, it is apparent that the CNC milled part has the better form error of the two parts, coming within 0.02 mm of the expected errors. The data taken from the part is also less noisy than the data from the SLA fabricated part, which can be physically seen from the differences in surface finish of the two parts. From this analysis, no information is given about the rotation or shift in the part. Using the fiducial analysis techniques developed in this work, these errors can be quantified and are shown in the following figure. The analysis for the CNC milled part is on the left, and the analysis for the SLA manufactured part is on the right. All units are in mm, and the first four digits past the decimal are significant.

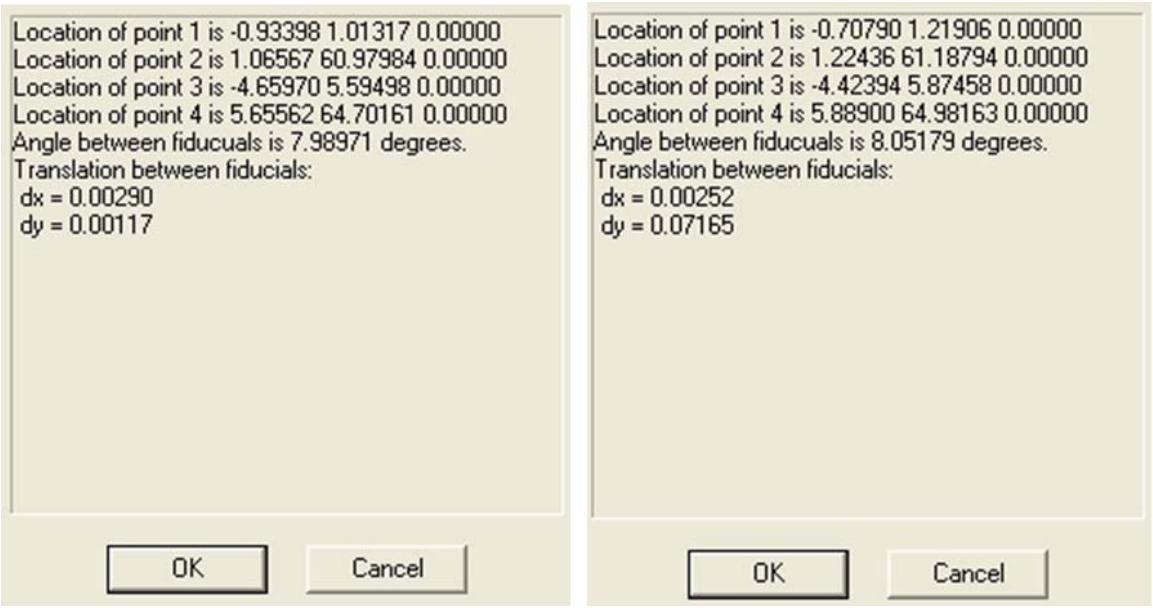

Figure 54: Results of fiducial analysis for rotated and scaled block

The theoretical rotation angle is 8.0000 degrees from the nominal geometry, and both test parts are within 0.05 degrees of that value. The theoretical shift should be zero in both x and y directions, and the results show good agreement. The CNC milled part is within 0.0029 mm in either direction, while the SLA made part contains only a 0.0025 mm shift in the x-direction but a larger, 0.0717 mm shift in the y-direction. However, the results for the SLA part are within the stated tolerances of the machine and are, thus, acceptable.

The last test parts contain both a shift and scale along the z-axis of the part. These parts are shown in the upper right of Figure 46 and Figure 47. The CAI scene after importing two digitized cross-sections in shown in Figure 55.

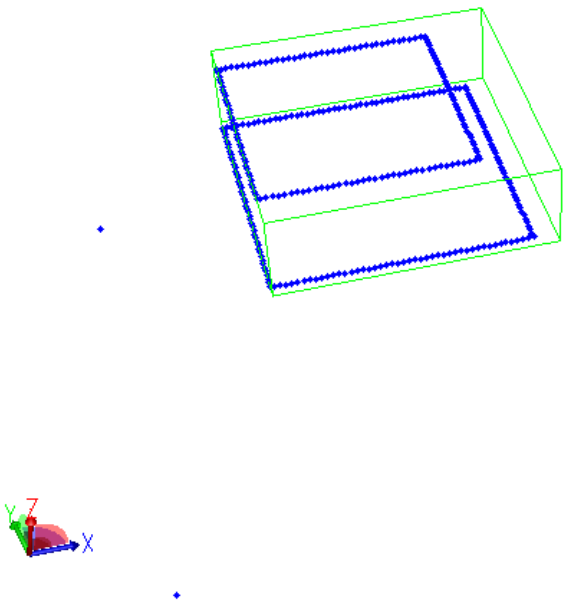

Figure 55: CAD model and point clouds with induced shift and scale error

Least-squares registration is first performed on each point cloud. The summary of the least-squares registration is shown in Table 18.

Table 18: Summary of results for rotated test part

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Theoretical Error - Top	-3.7250	0.0000	-3.7250	-3.7250
CNC Top	-3.7530	0.0505	-3.7192	-3.6970
SLA Top	-3.8594	0.0816	-3.9778	-3.4751
Theoretical Error - Bottom	-1.2750	0.0000	-1.2750	-1.2750
CNC Bottom	-1.2530	0.0025	-1.2663	-1.2472
SLA Bottom	-1.4286	0.0407	-1.4770	-1.1158

Again, the CNC milled part has the better form error of the two parts, coming within 0.033 mm of the expected errors. The data taken from the part is also less noisy than the data from the SLA fabricated part evidenced by the lower standard deviations. Using the fiducial analysis techniques developed in this work, the geometric errors between the two point clouds can be quantified and are shown in Figure 56. The analysis for the CNC milled part is on the left, and the analysis for the SLA manufactured part is on the right. All units are in mm, and the first four digits past the decimal are significant.

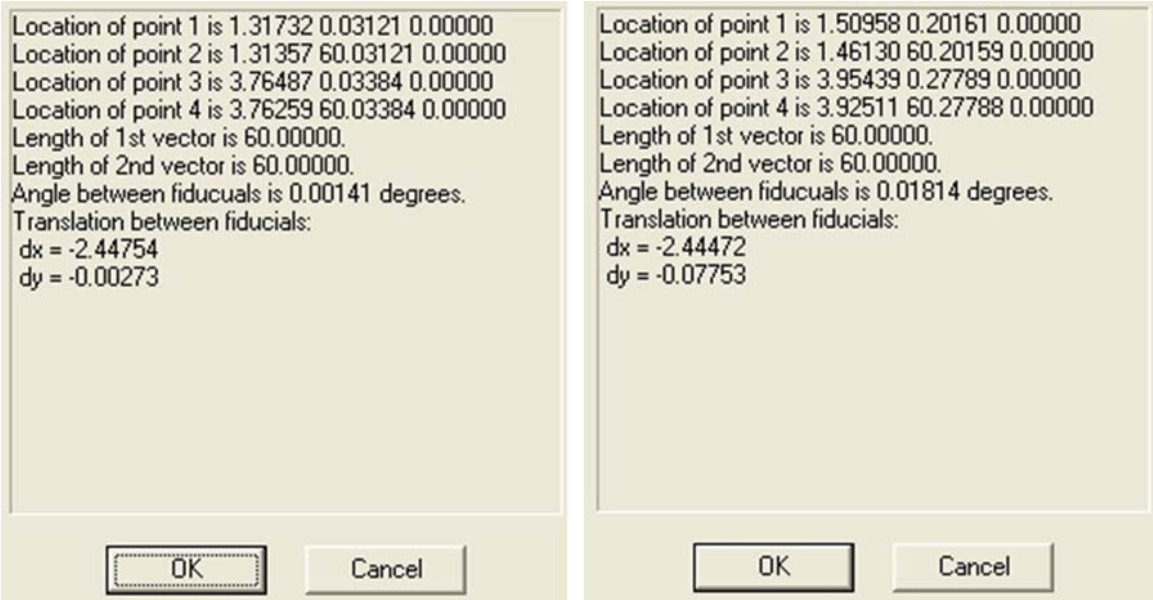

Figure 56: Results of fiducial analysis for shifted and scaled block

The theoretical shift is 2.4500 mm from the nominal geometry. The CNC milled part is within 0.0025 mm of that value, and the SLA part is within 0.0053 mm. Both results are within the stated tolerances of each respective machine. There should be no rotation between the two clouds, and both results show little rotation, with the CNC milled part again showing the better result.

LIGA Case Study

The last case study of this research focuses on the analysis of an actual LIGA fabricated part. The part of interest is a hollow rectangular block with dimensions of 2 x 3 x 1.5 mm. A CAD model of the part is shown in Figure 57.

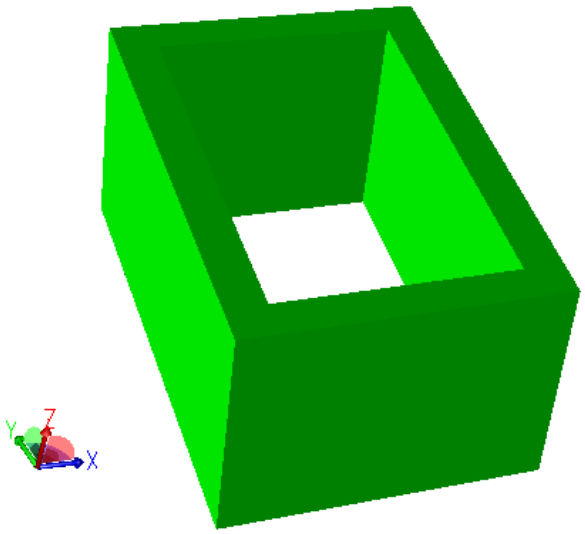

Figure 57: CAD model of inspected LIGA part

In order to inspect the LIGA part, a novel fixturing scheme was developed in coordination with Sandia National Laboratories. The metrology tool used to acquire edge data from the part is the View Engineering Voyager, described in Chapter III. In order to obtain data from both the top and bottom edges of the part, a fixturing scheme is devised to include fiducial markings which provide a reference between the top point cloud scan and the bottom point cloud scan. The proposed designed fixture is shown in the following figure.

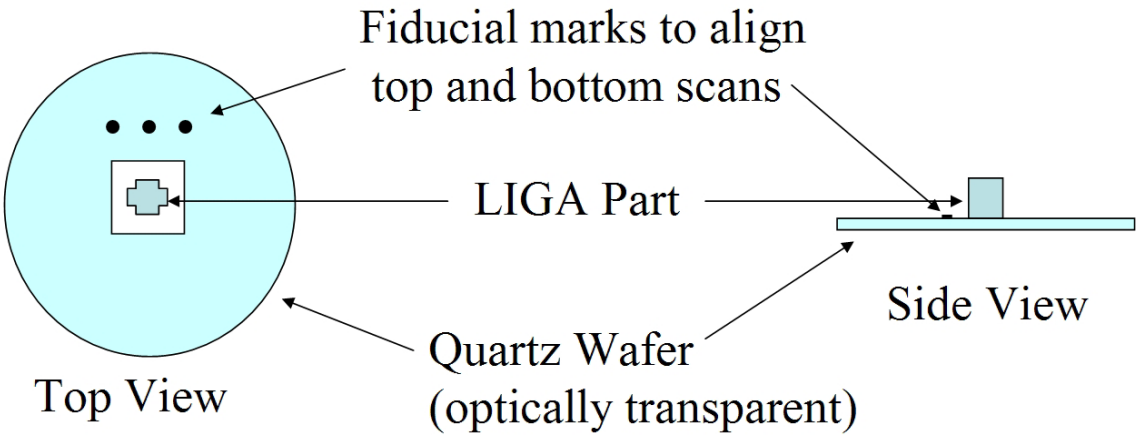

Figure 58: Fixture design for LIGA inspection

The actual fixture developed by Sandia National Laboratories is shown in Figure 59. The fixture consists of an optically transparent quartz wafer to mount the part. The quartz wafer is chosen for both its optical clarity as well as its parallelism. This particular wafer has a wedge error of less than 2 μm over a 127 mm square, which is extremely parallel. The quartz wafer is placed on another fixture which consists of three tooling balls with radii of 6.350 ± 0.006 mm. Placing the quartz wafer on these three ball bearings ensures parallelism with the inspection stage of the vision system. On the quartz wafer are small, chrome cylinders with nominal radii of 0.250 mm that are used as the fiducial markings. The LIGA part is mounted on the quartz wafer with an optically transparent adhesive wax.

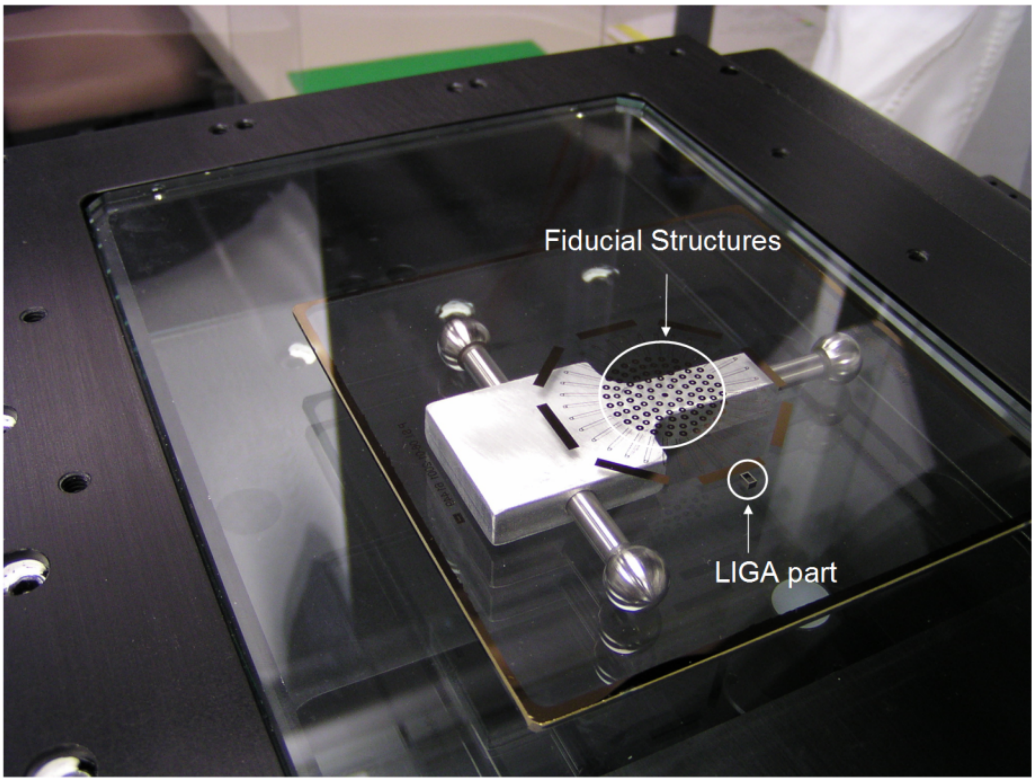

Figure 59: Fixture developed at Sandia National Laboratories

Using the View Voyager, scans of the tops edges of the LIGA part are made in conjunction with the top edges of the three fiducial marks. Once the top scan has been acquired, the entire fixture is rotated 180 degrees about the X-axis of the machine allowing the bottom surface of the part to be scanned. Because the mounting wafer is optically transparent, scans can be taken of the bottom edges of the LIGA part as well as the fiducial marks. Once information about the part has been digitized, the two point clouds and CAD model can be loaded into the developed software platform as shown in Figure 60.

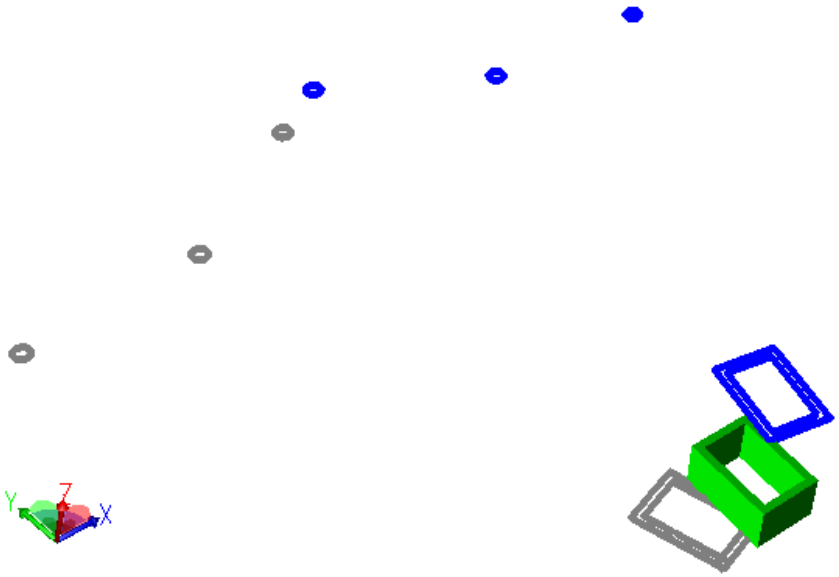

Figure 60: CAI scene of LIGA model and imported point clouds

Before any type of registration can be performed, the CAD model should be checked for the directions of the normals of the edges. The normals of the edges determine what points are "inside" or "outside" the designed CAD model. Since the CAD model consists of only linear geometry, what points are considered "inside" or "outside" is completely arbitrary. Figure 61 illustrates the normal directions of the edges from the imported CAD model. The sidewall geometry is excluded for the sake of clarity.

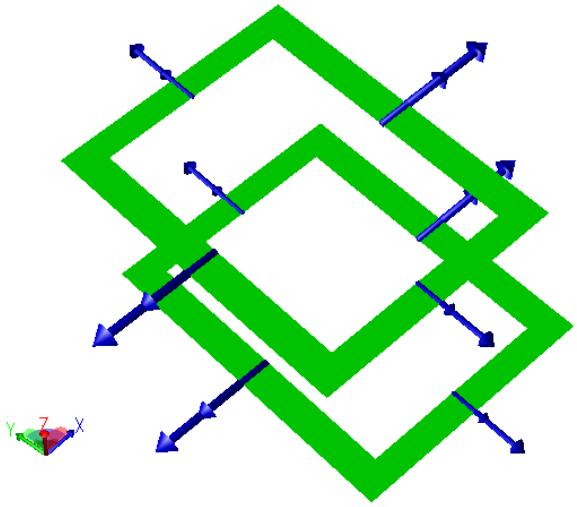

Figure 61: Normals of imported CAD model

From the imported CAD model, it is clear that the normals for the edges of the outside perimeter of the part are correct. The normals for the inside perimeter, however, are currently in the wrong direction. Any point on the "inside" of the part would be incorrectly represented as "outside" of the part, and metrics such as the mean of the errors would be misrepresented. The implemented software platform allows for the correction of this misrepresentation by allowing the user to manually change the direction of each normal. Figure 62 illustrates the normal directions of the edges after the correction. Again, the sidewall geometry is excluded for the sake of clarity.

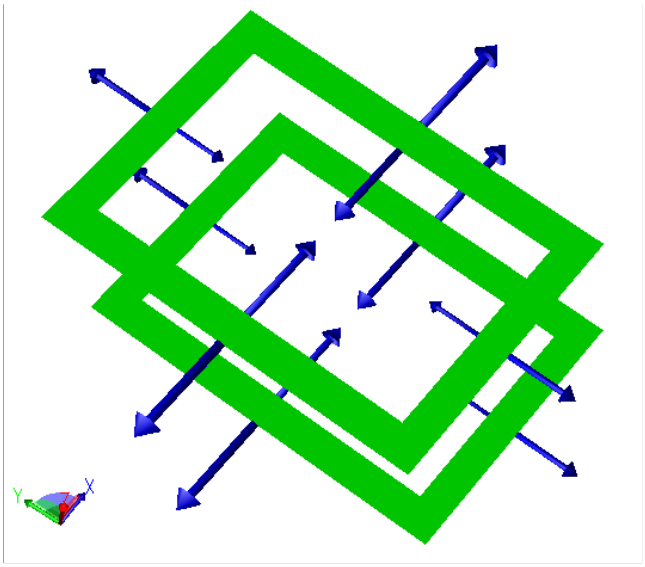

Figure 62: Corrected normal directions of edges

After the normals have been corrected, least-squares registration is performed for both the top and bottom point clouds. Because the data are planar, the transformations that apply to this case are x-translation, y-translation, and z-rotation. Since the fiducial data are in the same point cloud as the top or bottom scan, these points must be hidden as not to affect the registration process. Figure 63 shows the result of the least-squares registration of both the top and bottom point clouds.

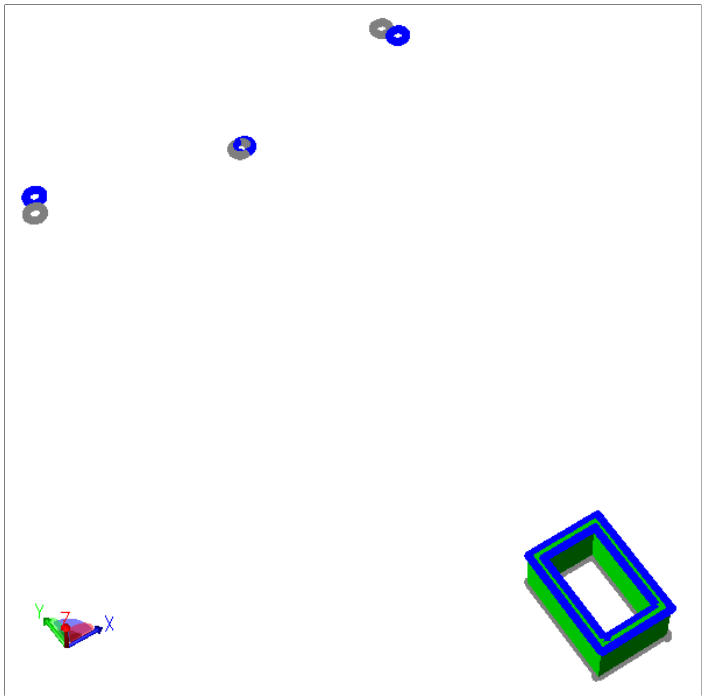

Figure 63: Point clouds after registration

Looking at the whisker plot of the results, a certain number of outliers are obvious in the point clouds. Outliers typically occur in data from vision systems because of erroneous "edges" detected from surface roughness or any other feature/defect which has significant contrast differences from the surrounding areas. Figure 64 illustrates some of the obvious outliers in the point cloud data for the bottom scan. The top point cloud has far fewer outliers (i.e. only three were found in the data).

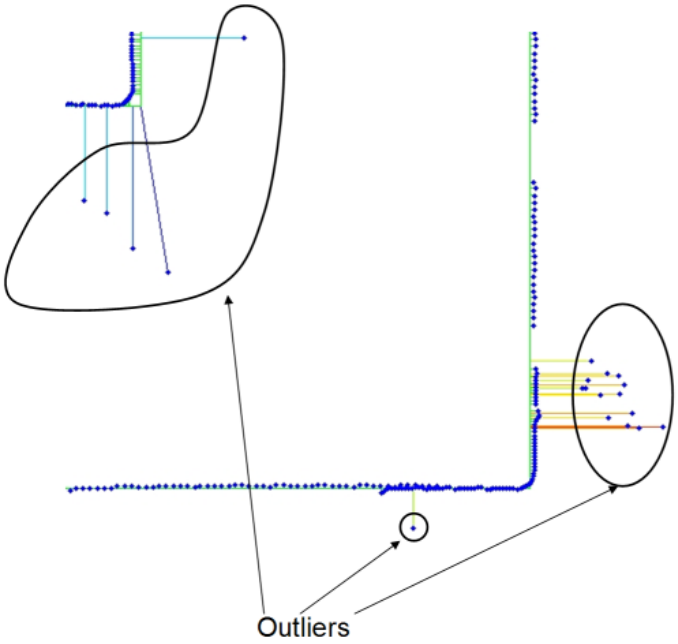

Figure 64: Outliers present in bottom point cloud

Least-square registration was performed with the raw data as well as with the outliers excluded from the analysis. The results from the registration process are summarized in Table 19. All results are in mm, reported to the resolution of the vision system.

Table 19: Summary of results for LIGA test part

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Bottom point cloud	0.0042	0.0061	-0.1101	0.0940
Bottom cloud – No Outliers	0.0040	0.0024	-0.0045	0.0115
Top point cloud	0.0040	0.0024	-0.0903	0.0089
Top cloud – No Outliers	0.0041	0.0017	-0.0013	0.0089

The results from the least-squares registration show that both the top and bottom are 4 μm oversized on average. It can be seen that the bottom point cloud standard deviation decreases significantly when the outliers are removed from the analysis. The effect is smaller on the top point cloud because there are few outliers to begin with. The means of the errors for the top and bottom agree to within 0.0001 mm, meaning there is little taper in the part. For a LIGA part with the given dimensions, the form error is acceptable.

Performing a fiducial analysis allows for geometric errors between the two point clouds to be quantified. Before performing a fiducial analysis, characterization of the fiducials should be performed. Because the fiducials are physically small cylinders, geometric parameter fitting is used to fit circles to the top and bottom of the cylinders. The centers of these circles serve as the fiducial points for the analysis. With the current fixture, there are three fiducial cylinders available for analysis. The fiducial analysis scheme developed in this research only requires two, and, thus, the third fiducial will be used to verify the results from the first two fiducials. The two properties that are calculated for the fiducials before performing the analysis are the radius of each circle and the z-height of each fiducial. Theoretically, the radius of each circle should not matter, as long as top and bottom circles of the fiducial are concentric with each other. Concentricity is assumed because with the small height of the cylinders, there is little chance for shift in the fiducial. The fiducials should also be planar with respect to each other, in order to accurately perform the fiducial analysis. Table 20 displays the measured dimensions of the top and bottom circles of the three fiducials.

Table 20: Summary of fiducial measurements

	Radius	Z-height
Circle 1 – Top	0.2489	0.0091
Circle 1 – Bottom	0.2488	0.0001
Circle 2 – Top	0.2489	0.0093
Circle 2 – Bottom	0.2489	0.0001
Circle 3 – Top	0.2488	0.0092
Circle 3 - Bottom	0.2489	0.0001

All of the radii measurements are within 0.0001 mm, which is well within the stated accuracy of the vision system. The Z-heights of the circles are within 0.0002 mm of each other, also within the stated accuracy of the machine. From the results, the three cylinders on the part can be deemed good candidates for the fiducial analysis given the current metrology hardware.

The fiducial analysis is performed with all combinations of the fiducials. Each fiducial is labeled as depicted in Figure 65. The point of rotation is taken to be about the centerline axis of the CAD model, also shown in Figure 65.

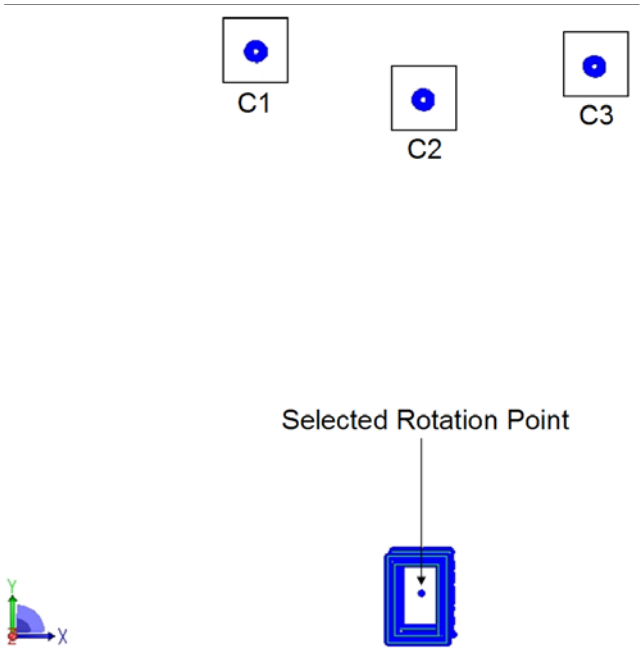

Figure 65: Fiducial labeling and rotation point

Performing the fiducial analysis with each combination of possible solutions, the following figure results: (all units are in mm and the first four digits past the decimal are significant)

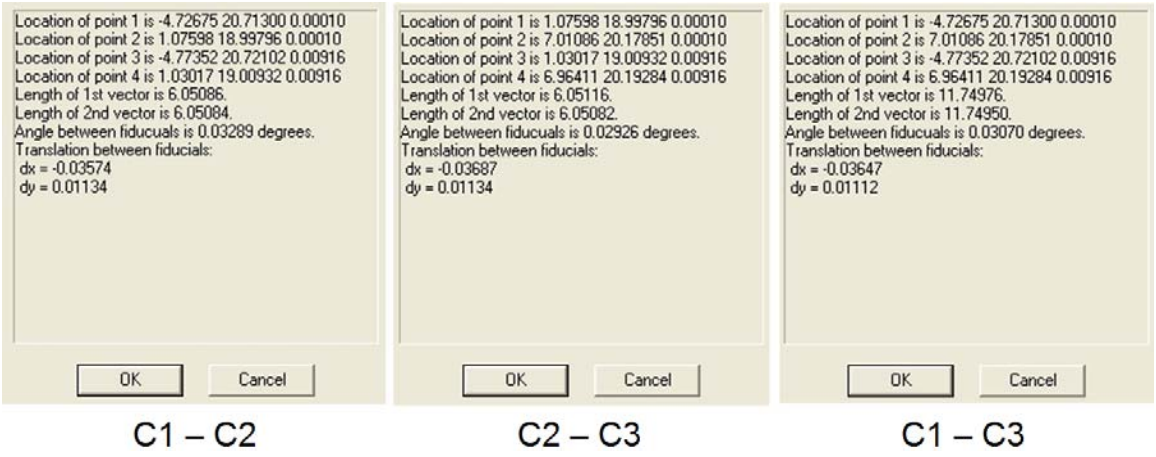

Figure 66: Results of fiducial analysis for LIGA part

The first metric of importance is the length of the vectors used for the fiducial analysis. The 1st vector length in each analysis represents the center-center distance between two circle fiducial marks of the bottom point cloud. The 2nd vector length represents the center-center distance between two circle fiducial marks of the top point cloud. These lengths should be identical in an ideal case. For all three cases, the difference in lengths is within 0.0003 mm for the 1st and 2nd vectors. The measurement uncertainty of the machine is 0.0035 mm, so these results are well within the machines specifications.

The results for twist about the centerline of the part are comparable between the three measurements. The average twist of the three measurements is 0.0309 degrees. The measurements of shift between the three results are also comparable, averaging -0.0364 mm in the x-axis and 0.0112 mm in the y-axis.

For the given part, these errors may or may not be acceptable depending on the application of the part which is unknown. This case study intends to illustrate the usefulness of the developed algorithms and shows the method in which a part can be inspected for both form errors of a single point cloud and geometric errors between point clouds.

CHAPTER VI

CONCLUSION AND RECOMMENDATIONS

Conclusions

This research serves to advance the goal of full geometric characterization of high aspect ratio MEMS. A critical link is established between point clouds acquired from the top and bottom of a MEMS device. A methodology is developed that allows for the characterization of gross geometric errors in a part that were previously unattainable.

The framework developed in this work uses previous research done in the metrology group at the Georgia Institute of Technology as the foundation for point cloud-to-CAD analysis. All previous work has been directed towards algorithms applicable to only three-dimensional geometry. This work adds significant extensions to the previous work, and results in a framework that is a superset of the previous work by allowing for both two-dimensional and three-dimensional analyses in a single environment. The derivations of these extensions for two-dimensional geometry are necessary in order to provide a method to analyze data from MEMS inspection hardware.

The main contribution of this work, however, is a new methodology developed specifically for multiple, two-dimensional point cloud analysis. The developed 2.5-D inspection technique allows for the characterization of both the form errors and geometric errors in high aspect ratio parts.

With the current metrology hardware available for high aspect ratio MEMS, full three-dimensional geometric characterization of parts is not possible. The best current alternative for high aspect ratio MEMS inspection is to gather two-dimensional scans of the tops and bottoms of the part, and apply the methods developed in this work. The developed methodology uses multiple two-dimensional point clouds that include fiducial information and implements a quantitative analysis algorithm that provides information about the overall part geometry. Applying the methodology developed in this work provides both form error and overall geometric error information for a given part. The methodology of using fiducial information to characterize three-dimensional errors in high aspect ratio parts from only two-dimensional data is a significant contribution to the field of MEMS metrology. In combination with the developed fiducial fixture, the methodology serves as a novel technique for the inspection of high aspect ratio MEMS. New information is provided about the overall geometry of these parts that was previously unobtainable.

The simulated tests prove the developed algorithms are accurate and repeatable. The analysis of the two sets of test parts manufactured by the CNC mill and SLA result in the expected outcome: that the CNC-machined parts have the tighter tolerances and better accuracy. More importantly, the methodology developed is shown to be applicable

to physical parts and data acquired by metrology hardware, and it produces the expected results given parts with known errors induced.

The LIGA test case gives some insight into the fabrication process for this particular part. Looking at the residual errors after least-squares registration, the form of the top and bottom geometry are shown to be accurate, with the means of the error for both the top and bottom being 4 μm oversized. Depending on the function of this part, the results are more than likely an acceptable form tolerance. The fiducial analysis shows a shift between the top and bottom with the more significant error being in the X-direction of the model coordinate system. This error is computed to be 36 μm over 1500 μm, or 2.4% over the height of the part. A possible explanation for the acceptable form errors, but significant geometric errors, is that the X-rays were not orthogonal to the substrate during the exposure step of the process. Investigation into other causes for this measured error is ongoing at Sandia National Laboratories.

Contributions

This work includes a number of intellectual contributions to both the fields of MEMS metrology and CAI. The major contributions are as follows:

- Provides the next step towards full three-dimensional geometric characterization of high aspect ratio MEMS by establishing a methodology for 2.5-D analysis using multiple point clouds. Current inspection practices only use 2-D analyses to quantify a part. The

developed methodology allows for the calculation of both form errors and geometric errors based on two-dimensional scans of a MEMS device;

- Develops the mathematical formulations of a four-point fiducial analysis that is used to quantify errors between multiple point clouds;
- Discovers possible errors associated with deviations calculated to finite geometries, and develops the mathematical derivations to efficiently correct these errors for two-dimensional geometry;
- Proposes a novel inspection fixture design that provides fiducial information for top and bottom scans of high aspect ratio MEMS;
- Implements a framework resulting in a software platform for the analysis of both 2-D and 3-D CAD models and coordinate data, irrespective of the data acquisition source.

Recommendations for Future Work

The field of MEMS metrology is still in its infancy. Much research is ongoing to innovate and improve current data acquisition techniques for MEMS devices. Eventually, a method will exist that allows for the acquisition of a full three-dimensional point cloud of a MEMS device.

One suggested method to further advance the geometric characterization of MEMS devices is to develop a fixturing scheme for a MEMS device that allows fiducial information to be measured for all surfaces and edges of the part. In this research, a fixture setup is presented that allows for top and bottom scans of a part to be correlated;

however, a fixture to inspect all sides of a part requires more extensive research. For macro scale parts, methods exist that allow for multiple data scans to be connected using known geometry between the scans. An analogous fixturing scheme for MEMS will enable for the combination of multiple point clouds of a MEMS part into a full three-dimensional point cloud. An extension of the fiducial setup developed for this research will enable scans of the sidewalls of a part to be combined with scans of the top and bottom of the part. This accomplishment would be another significant step towards full part MEMS inspection. If a fixture could be designed that will enable for the inspection (not necessarily limited to the vision system used in this research) of all surfaces of a part as well as fiducials common to all individual point clouds, new information could be inferred about the geometric characteristics of a MEMS part.

With respect to the body of knowledge for coordinate data analysis, methods to optimize deviation calculations for three-dimensional geometry are suggested. In this work, it was discovered that noisy data could be improperly assigned to certain geometries based on the definition of geometry. Since two-dimensional geometry is the primary geometry used in MEMS design, analytic derivations were devised in this research to solve the problem of trimmed geometry and improper assignment. Deriving analytical solutions to solve trimmed geometry deviations for three-dimensional geometry would be a significant contribution for general CAI cases, MEMS related or not. The work done in this research solves the problem of three-dimensional trimmed geometry with the use of lower level modeling kernel function calls. If analytical formulations could be derived for the case of trimmed three-dimensional geometry, the

function calls to the modeling kernel which add computational overhead could be avoided.

APPENDIX A

EXPERIMENTAL VALIDATION

Overview

This appendix shows the results for the deviation verification calculations for all geometry types except straight lines, which were discussed in Chapter V. The methodology to verify each geometry deviation calculation is exactly the same as the method outlined in Chapter V. For each geometry, a paragraph explains the test CAD model and the appropriate transformations that apply to the model. Following that information, a series of tables and figures are presented which correspond to those for straight line geometry in Chapter V. Much of the repetitive descriptive text is left out for conciseness. The overall result is that each deviation calculation for each geometry type is shown to be robust and accurate, both with and without the presence of noise.

Circular Curve

A CAD model of a circular face is modeled in order to verify the deviation calculations for circular geometry. This face contains one circular edge which will be used for the circle deviation calculation. A point cloud consisting of one thousand points is generated to lie exactly (to the precision of the machine) on the circle. Just as in the straight line case, the registration routine should find the best fit transforms and output deviations equal to zero for each point. The transform parameters that apply to this

scenario are x-translation, y-translation. Since the CAD geometry is circular, a rotation component is excluded for the deviation computation because it has no effect on the deviation calculations. In fact, if a rotation component is allowed, the minimization step may become unstable because the minimizer may not find an optimal value for the rotation component. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithm for circular geometry. The procedure is similar to that of the analysis of straight line geometry. Table 21 lists the starting positions of each point cloud before registration. Figure 67 graphically shows each transformed point cloud and the final registered location.

Table 21: Initial starting positions before registration for circular geometry

	X translation (linear)	Y translation (linear)
Optimal Fit	**0.0**	**0.0**
Initial Guess A	0.015	0.015
Initial Guess B	0.45	0.3
Initial Guess C	0.5	-0.3
Initial Guess D	-0.3	0.4
Initial Guess E	-0.2	-0.4

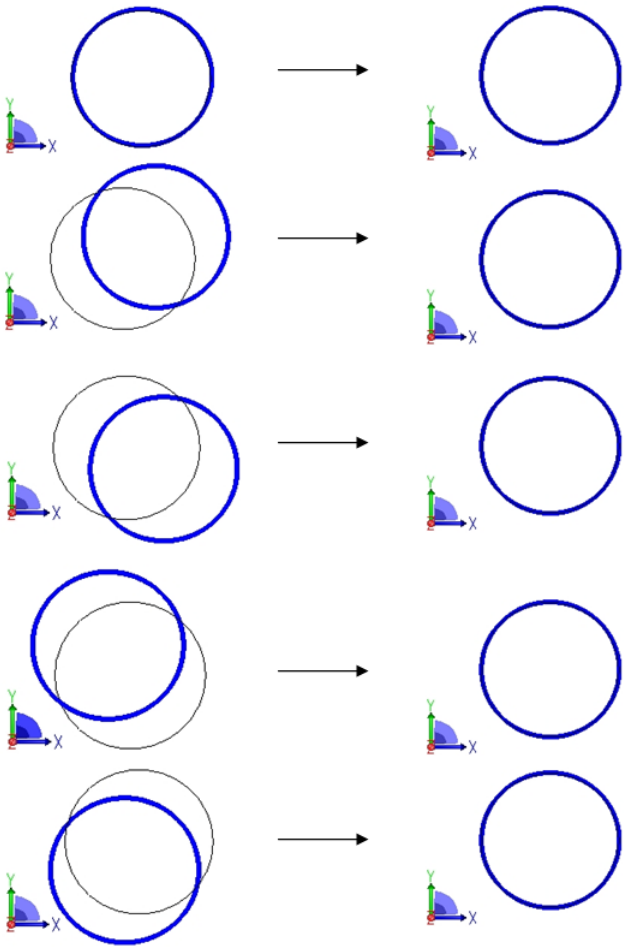

Figure 67: Initial guesses and registered point clouds for circular geometry

Table 22: Deviation summary for test cases for circular geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	-0.000000	0.000000	-0.000000	0.000000
Initial Guess B	-0.000000	0.000000	-0.000000	0.000000
Initial Guess C	-0.000000	0.000000	-0.000000	0.000000
Initial Guess D	-0.000000	0.000000	-0.000000	0.000000
Initial Guess E	-0.000000	0.000000	-0.000000	0.000000

Table 23: Deviation summary for 0.001 unit noise for circular geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000008	0.000996	-0.002700	0.003556
Initial Guess A	0.000008	0.000995	-0.002679	0.003556
Initial Guess B	0.000008	0.000995	-0.002679	0.003556
Initial Guess C	0.000008	0.000995	-0.002679	0.003556
Initial Guess D	0.000008	0.000995	-0.002679	0.003556
Initial Guess E	0.000008	0.000995	-0.002679	0.003556

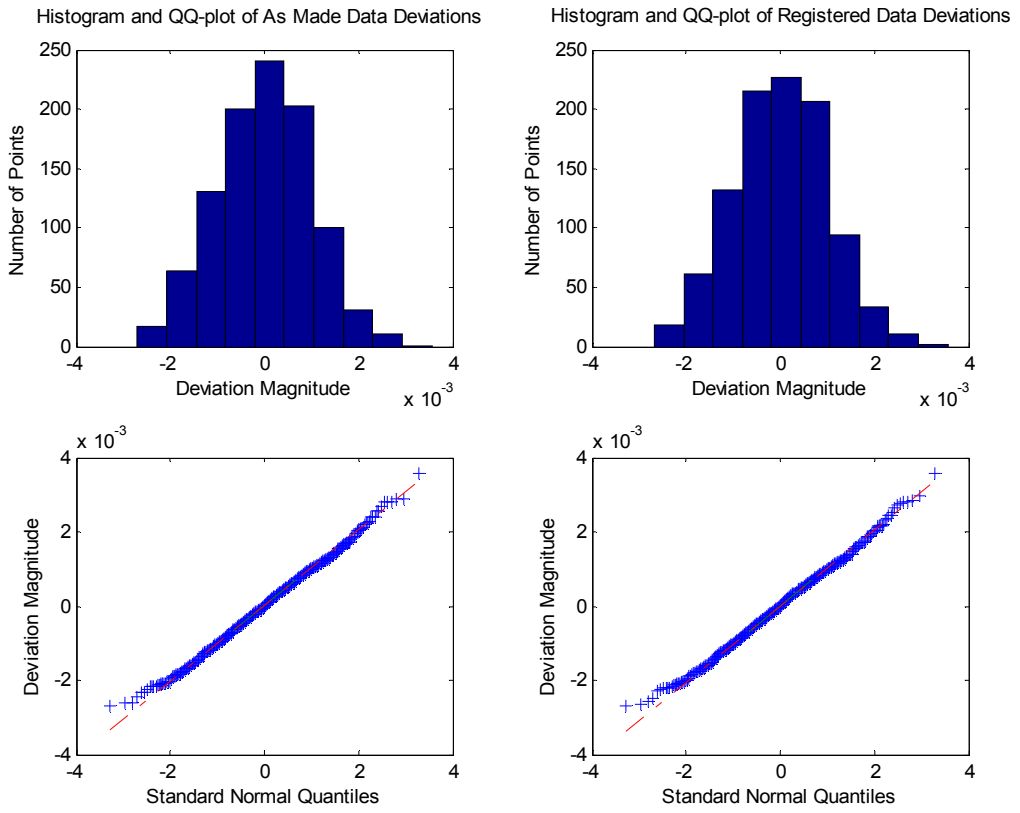

Figure 68: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for circular geometry

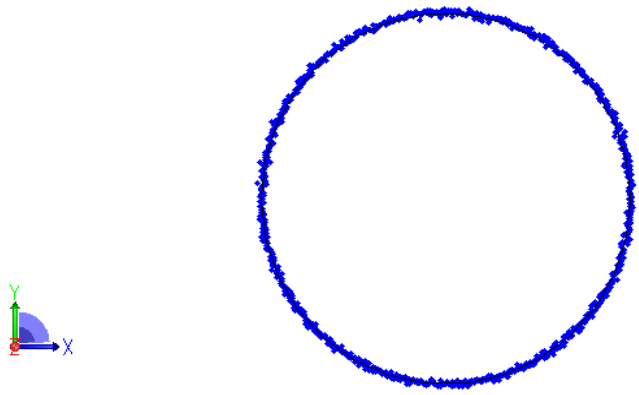

Figure 69: Data set with 0.01 units of noise for circular geometry

Table 24: Deviation summary for 0.01 unit noise

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000624	0.010123	-0.030876	0.032288
Initial Guess A	0.000624	0.010114	-0.030514	0.032821
Initial Guess B	0.000624	0.010114	-0.030514	0.032821
Initial Guess C	0.000624	0.010114	-0.030514	0.032820
Initial Guess D	0.000624	0.010114	-0.030514	0.032821
Initial Guess E	0.000624	0.010114	-0.030514	0.032820

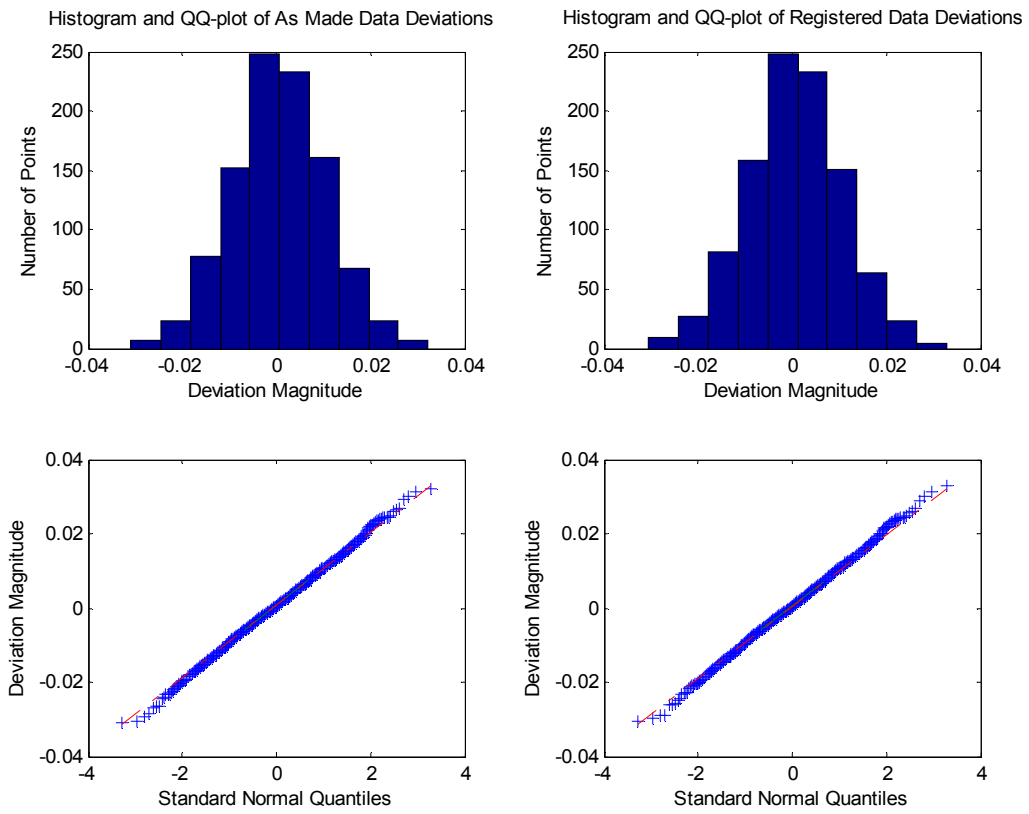

Figure 70: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for circular geometry

Since there is only one circular geometric entity in this CAD model, the occurrence of improperly assigned points is not possible. In order to demonstrate possible errors with circular geometry and noisy data, another CAD model was generated as shown in Figure 71. This CAD model represents a half cylinder with one circular arc and one linear edge.

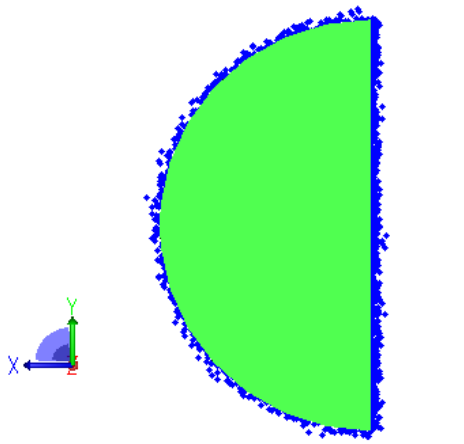

Figure 71: CAD model for trimmed geometry analysis for circular geometry

Using the standard deviation calculation derived in Equation (6) and a data set with 0.01 units of noise, the registration of the point cloud to the CAD model shows errors in the deviation calculations. Figure 72 shows the top intersection of the circle and line and the points that are improperly assigned.

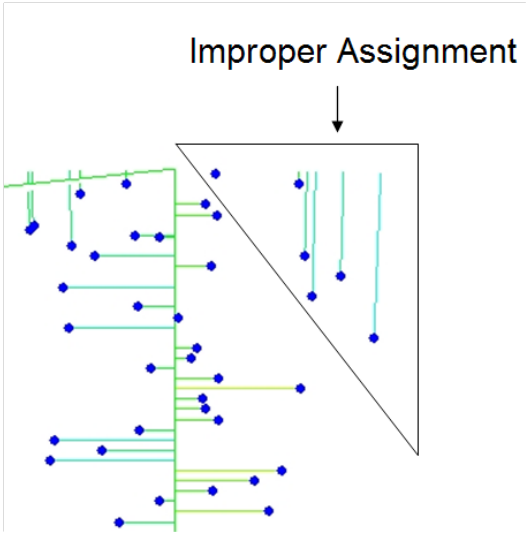

Figure 72: Improper assignment of points to underlying CAD geometry

At the intersection of the circular arc and line segment of the CAD model, some of the points are assigned to circular geometry that is not present. It should be noted that the above whisker plot does not appear to correctly draw the whiskers to the nominal CAD geometry for parts of the circular arc. This observation is due to the faceting of the geometry for the visualization kernel and is not an error in the calculations.

The reason for the improperly assigned points is similar to the reason for the straight line case. In order to account for this type of error, the closest point on the circle must be calculated using the derivations in Equations (30) - (33). Again, these calculations impose a greater computation penalty on the deviation calculation, but if the data are sufficiently noisy, incorporating trimmed geometry may be the most reliable technique for registration. Figure 73 shows the partial whisker plot using the trimmed deviation calculations for circular geometry. Again, the whiskers which appear to not lie on the actual circular geometry are due to the faceting of the CAD model for visualization, not errors in the algorithm.

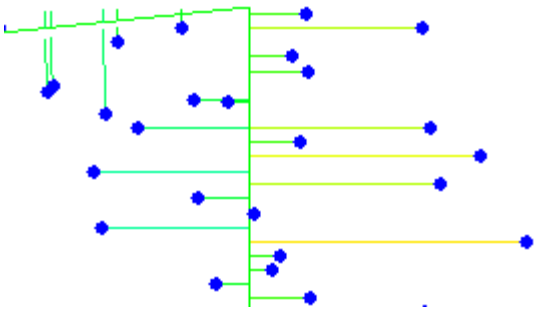

Figure 73: Proper assignment of points to underlying trimmed geometry

General Parametric Curve

In order to verify the deviation calculations for a general parametric curve, a closed spline CAD model is developed. This face contains one spline edge which consists of a series of composite third order non-uniform B-splines. A point cloud consisting of one thousand points is generated to lie exactly (to the precision of the machine) on the spline curve. The transform parameters that apply to this scenario are x-translation, y-translation, and z-rotation. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithm for general parametric geometry. The procedure is similar to the analysis of straight line geometry. Table 25 lists the starting positions of each point cloud before registration. Figure 74 graphically shows each transformed point cloud and the final registered location.

Table 25: Initial starting positions before registration for parametric geometry

	X translation (linear)	Y translation (linear)	Z rotation (radians)
Optimal Fit	**0.0**	**0.0**	**0.0**
Initial Guess A	-0.01	0.01	-0.01
Initial Guess B	0.45	0.5	0.2
Initial Guess C	0.5	-0.25	-0.1
Initial Guess D	-0.4	0.45	0.3
Initial Guess E	-0.25	-0.45	0.35

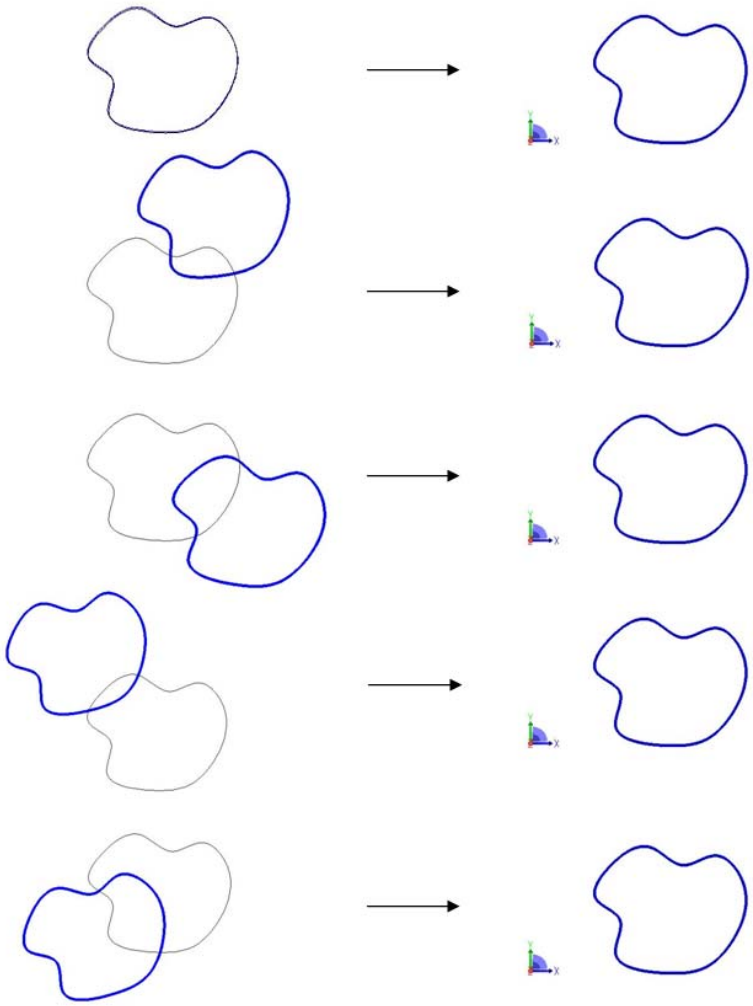

Figure 74: Initial guesses and registered point clouds for parametric curve geometry

Table 26: Deviation summary for test cases for parametric geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	0.000000	0.000000	-0.000000	0.000000
Initial Guess B	0.000000	0.000000	-0.000000	0.000000
Initial Guess C	0.000000	0.000000	-0.000000	0.000000
Initial Guess D	0.000000	0.000000	-0.000000	0.000000
Initial Guess E	0.000000	0.000000	-0.000000	0.000000

Table 27: Deviation summary for 0.001 unit noise for parametric geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made	-0.000005	0.000999	-0.003256	0.003334
Initial Guess A	-0.000005	0.000998	-0.003269	0.003359
Initial Guess B	-0.000005	0.000998	-0.003269	0.003359
Initial Guess C	-0.000005	0.000998	-0.003269	0.003359
Initial Guess D	-0.000005	0.000998	-0.003269	0.003359
Initial Guess E	-0.000005	0.000998	-0.003269	0.003359

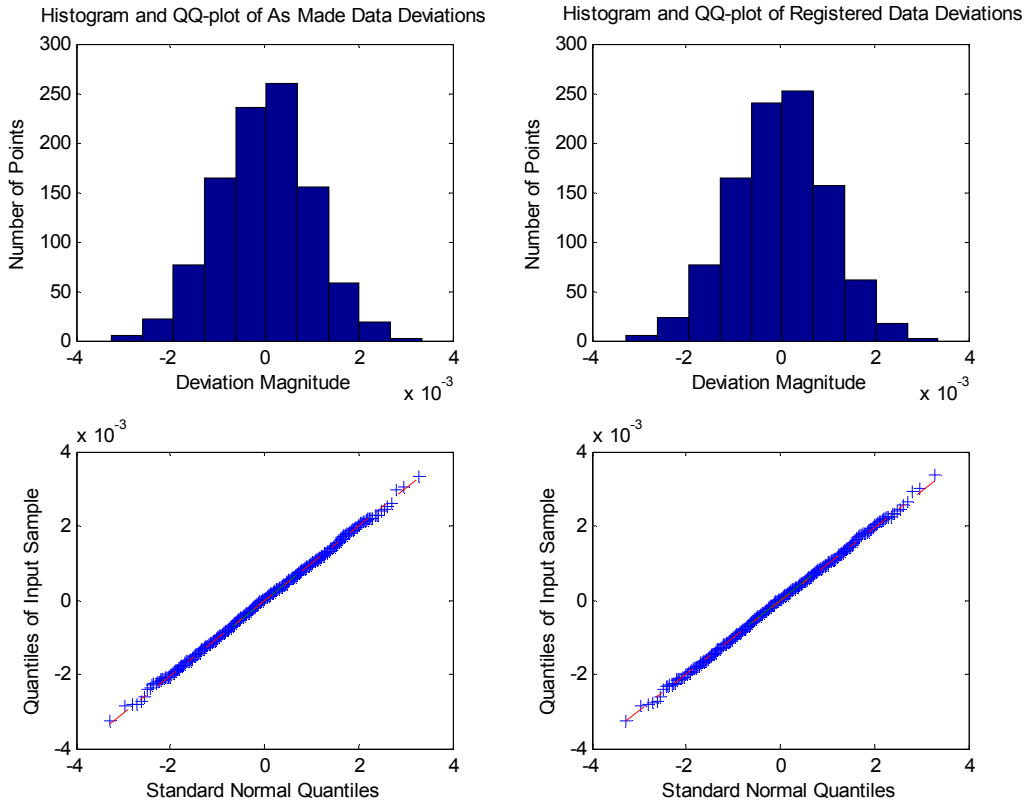

Figure 75: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for parametric geometry

Table 28: Deviation summary for 0.01 unit noise for parametric geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made	0.000377	0.010081	-0.030327	0.034908
Initial Guess A	0.000377	0.010073	-0.029972	0.034689
Initial Guess B	0.000377	0.010073	-0.029972	0.034689
Initial Guess C	0.000377	0.010073	-0.029972	0.034689
Initial Guess D	0.000377	0.010073	-0.029972	0.034689
Initial Guess E	0.000377	0.010073	-0.029972	0.034689

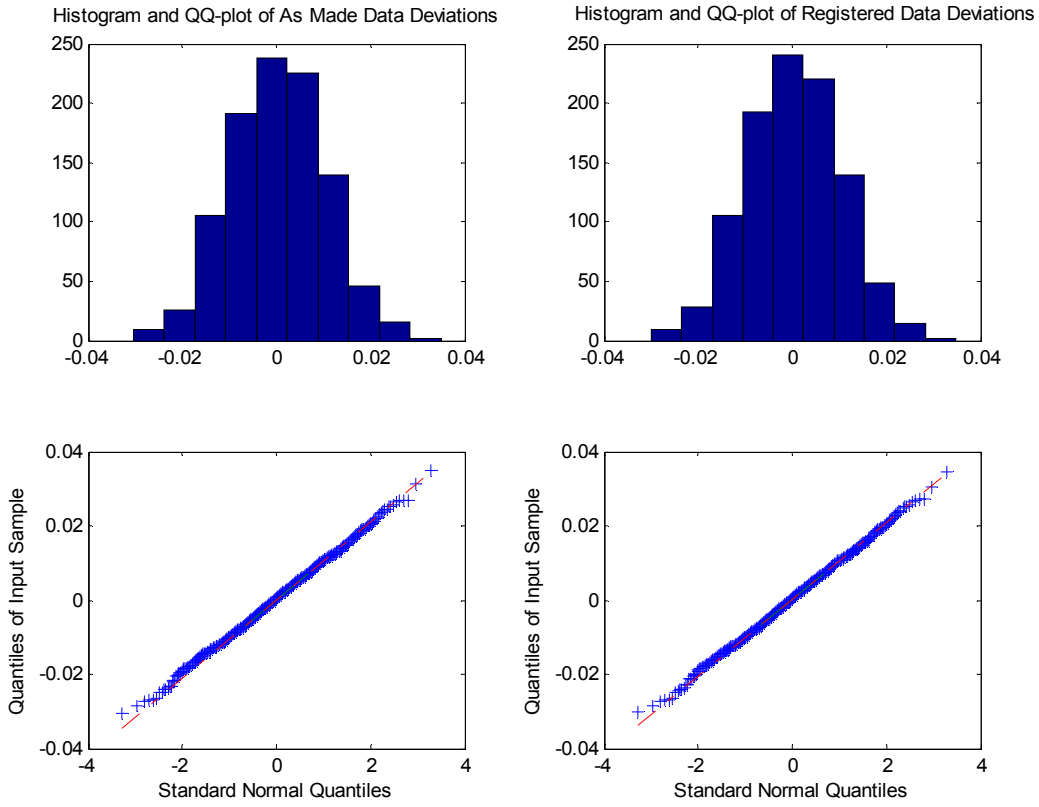

Figure 76: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for parametric geometry

Since lower level calls to the geometry kernel are used to calculate the deviations to general parametric geometry, the closest-point containment problem is solved using the kernel. Therefore, improperly assigned points should not be a factor. As mentioned previously, there is a significant performance penalty associated with these calls, and CAD models with analytic definitions should be used whenever possible for both accuracy and speed.

Plane

A CAD model of a planar face is modeled in order to verify the deviation calculations for plane geometry. This CAD model contains one planar face which will be used for the plane deviation calculations. A point cloud consisting of one thousand points is generated to lie exactly (to the precision of the machine) on the plane. The transform parameters that apply to this scenario are z-translation, x-rotation, and y-rotation. Since the underlying geometry is a plane, x-translation and y-translation do not affect the deviation calculations. Also, because of the definition used for the plane, z-rotation is not a factor in the deviation calculation. Registration to a trimmed plane is a separate case that will be explored at the end of this section. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithm for planar geometry. The procedure is similar to the analysis of straight line geometry discussed earlier. Table 29 lists the starting positions of each point cloud before registration. Figure 77 graphically shows each transformed point cloud and the final registered location.

Table 29: Initial starting positions before registration for planar geometry

	Z translation (linear)	X rotation (linear)	Y rotation (radians)
Optimal Fit	**0.0**	**0.0**	**0.0**
Initial Guess A	0.02	0.02	-0.01
Initial Guess B	0.2	0.3	0.2
Initial Guess C	0.4	-0.25	-0.1
Initial Guess D	-0.4	0.35	0.3
Initial Guess E	-0.25	-0.45	0.25

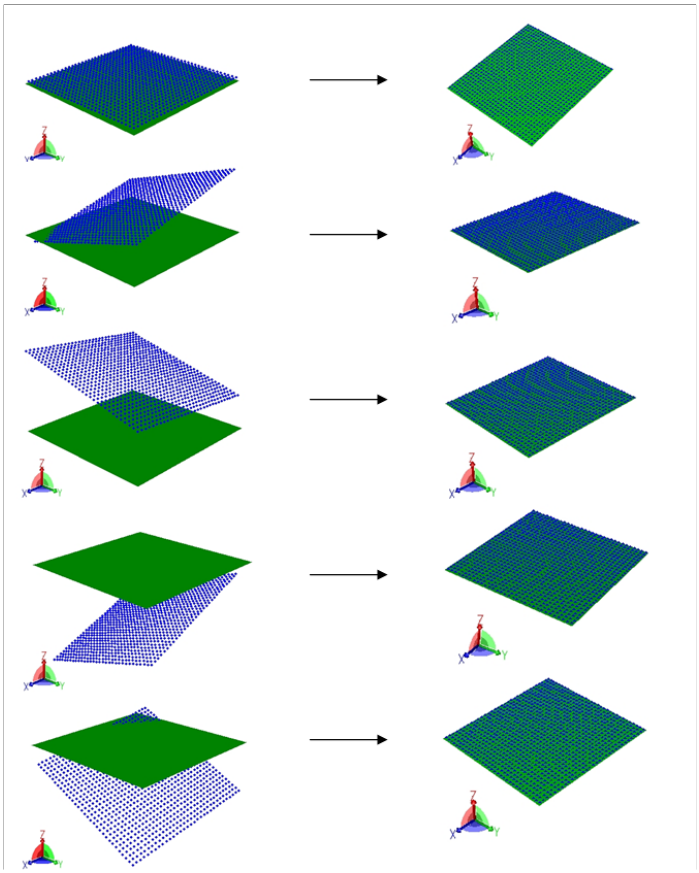

Figure 77: Initial guesses and registered point clouds for planar geometry

Table 30: Deviation summary for test cases for planar geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	-0.000000	0.000000	-0.000000	0.000000
Initial Guess B	-0.000000	0.000000	-0.000000	0.000000
Initial Guess C	0.000000	0.000000	-0.000000	0.000000
Initial Guess D	0.000000	0.000000	-0.000000	0.000000
Initial Guess E	0.000000	0.000000	-0.000000	0.000000

Table 31: Deviation summary for 0.001 unit noise for planar geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made	0.000029	0.000983	-0.003675	0.002910
Initial Guess A	-0.000000	0.000982	-0.003694	0.002904
Initial Guess B	-0.000000	0.000982	-0.003694	0.002904
Initial Guess C	-0.000000	0.000982	-0.003694	0.002904
Initial Guess D	-0.000000	0.000982	-0.003694	0.002904
Initial Guess E	-0.000000	0.000982	-0.003694	0.002904

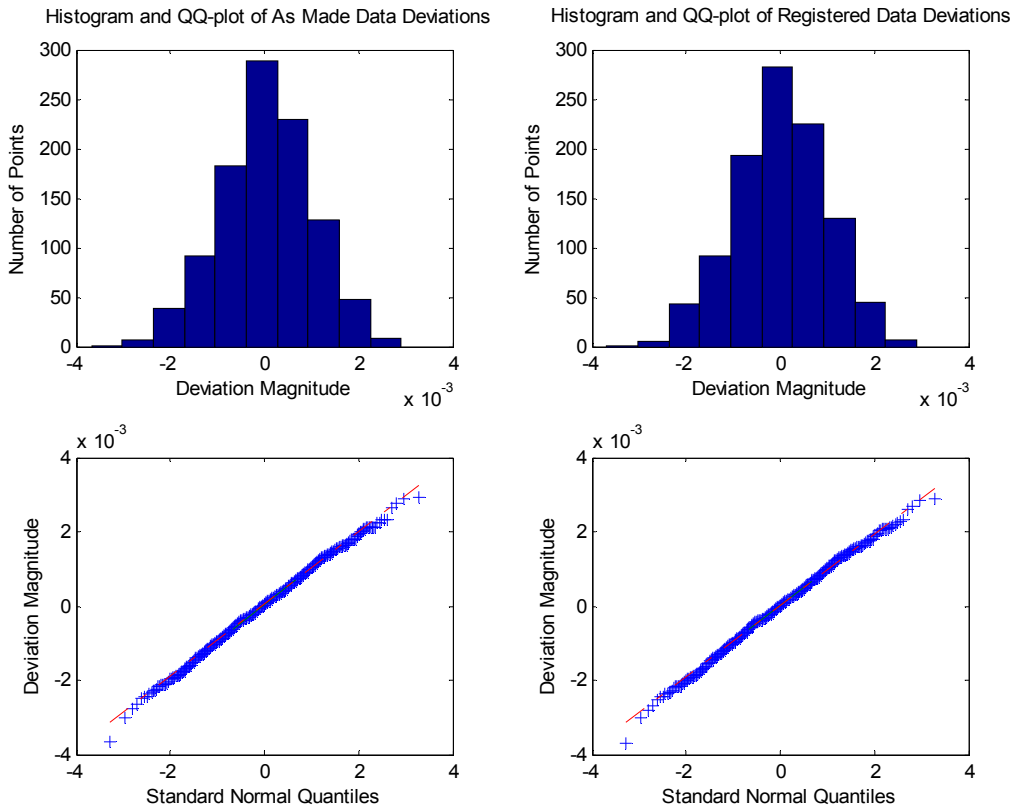

Figure 78: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for planar geometry

Table 32: Deviation summary for 0.01 unit noise for planar geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made	-0.000154	0.010063	-0.034235	0.029665
Initial Guess A	0.000000	0.010061	-0.034357	0.029974
Initial Guess B	0.000000	0.010061	-0.034357	0.029974
Initial Guess C	0.000000	0.010061	-0.034357	0.029974
Initial Guess D	0.000000	0.010061	-0.034357	0.029974
Initial Guess E	0.000000	0.010061	-0.034357	0.029974

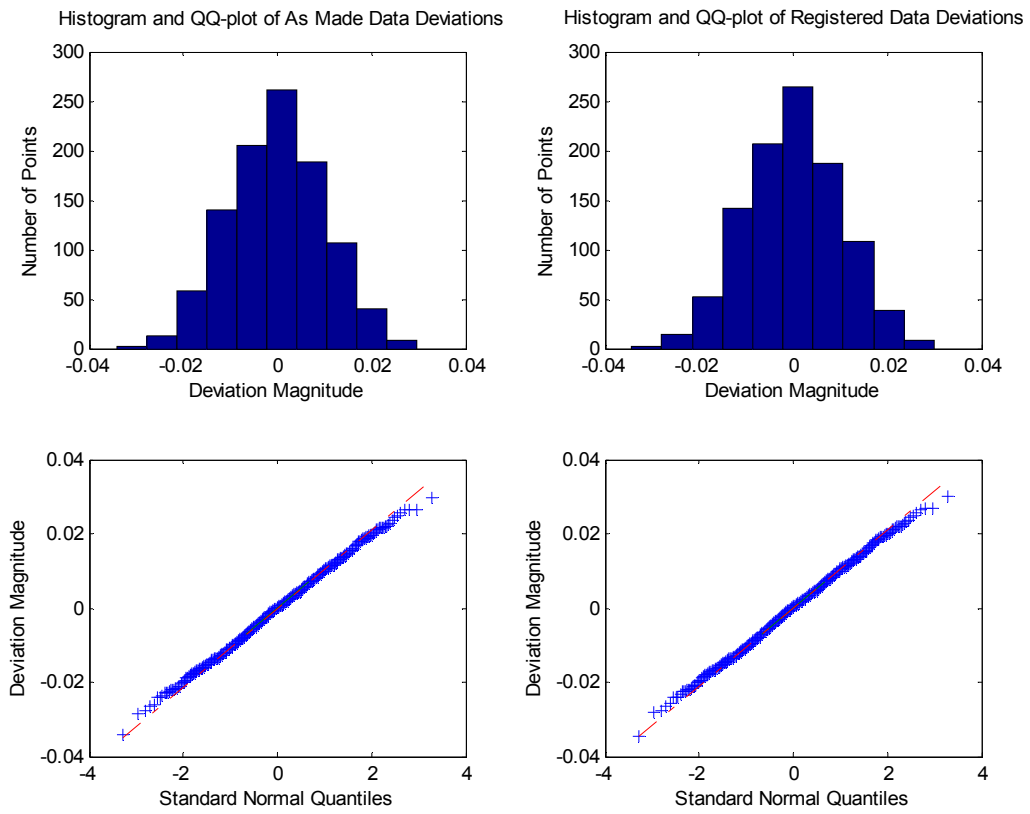

Figure 79: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for planar geometry

Fitting a trimmed plane is a special case for registration. In addition to computing the deviation from a point to a plane, the registration algorithm must also check to make sure the closest point normal to the point in question actually lies on the face of the CAD model. In the previous examples translation in x and y and rotation in z were not allowed because these transforms have no effect on the deviation calculations for a single plane definition (assuming the plane lies in the global XY-plane). In all practicality, the situation of registering to a CAD model consisting of a single planar face is unlikely because a plane will typically be one of many faces of a CAD model. For the sake of thoroughness, however, calculating deviations for trimmed planes is provided by making calls to lower level modeling functions. A hypothetical situation is shown in Figure 80.

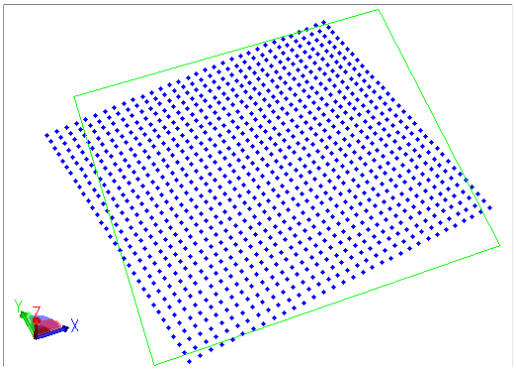

Figure 80: Fitting of trimmed plane

This particular example contains a point cloud, with no induced noise, which is rotated about the z-axis. If the data is registered to the CAD model without checking for the bounds of the plane, the point cloud is registered as shown in Figure 81.

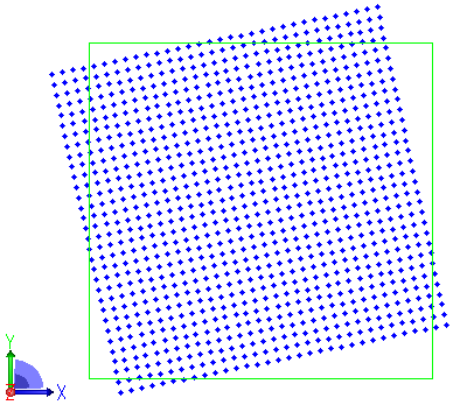

Figure 81: Registration without checking plane bounds

The deviations are minimized to zero in this particular case, but the rotation component is not found, simply because the geometric definition of the CAD model is a plane which, theoretically, is infinite in the x and y directions. Using an algorithm that checks for the bounds of the plane results in the following figure:

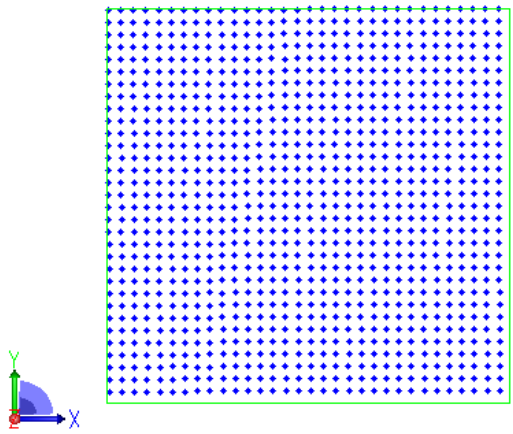

Figure 82: Registration with bounds checking

This result would probably be more desirable if this were an actual application. For three-dimensional geometry, determining if the closest point to a data point is actually on the face of the underlying geometry is beyond the scope of this work. Lower level calls to the modeling kernel are used to determine if the normal deviation to a surface actually lies on the CAD face. These calls to the modeling kernel are provided for each of the remaining three-dimensional deviation calculations, and the results are similar for the above planar case. Hence, the remaining deviation verifications will only apply to the algorithms developed in this work, and not to any of the lower level geometric modeling kernel function calls.

Sphere

A CAD model of a sphere is modeled in order to verify the deviation calculations for spherical geometry. This CAD model contains one spherical face which will be used for the sphere deviation calculations. A point cloud consisting of one thousand points is generated to lie exactly (to the precision of the machine) on the sphere. For this type of geometry, rotation transformations are unnecessary due to symmetry of the part. Five

different initial starting points are devised in order to test the repeatability and robustness of the algorithm for spherical geometry. The procedure is similar as those previously discussed. Table 33 lists the starting positions of each point cloud before registration. Figure 83 graphically shows each transformed point cloud and the final registered location.

Table 33: Initial starting positions before registration for spherical geometry

	X translation (linear)	Y translation (linear)	Z translation (linear)
Optimal Fit	**0.0**	**0.0**	**0.0**
Initial Guess A	-0.03	-0.02	-0.04
Initial Guess B	0.45	0.5	0.4
Initial Guess C	0.5	-0.55	-0.3
Initial Guess D	-0.45	0.35	0.3
Initial Guess E	-0.3	-0.45	0.35

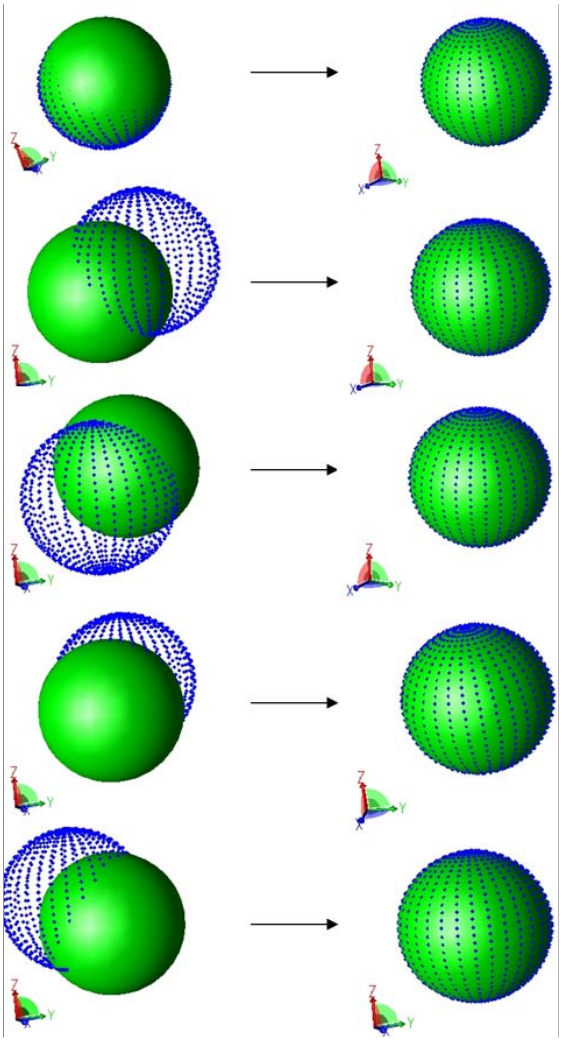

Figure 83: Initial guesses and registered point clouds for spherical geometry

Table 34: Deviation summary for test cases for spherical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	-0.000000	0.000000	-0.000000	0.000000
Initial Guess B	-0.000000	0.000000	-0.000000	0.000000
Initial Guess C	-0.000000	0.000000	-0.000000	0.000000
Initial Guess D	-0.000000	0.000000	-0.000000	0.000000
Initial Guess E	-0.000000	0.000000	-0.000000	0.000000

Table 35: Deviation summary for 0.001 unit noise for spherical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000016	0.000100	-0.002992	0.002825
Initial Guess A	0.000016	0.000100	-0.003020	0.002860
Initial Guess B	0.000016	0.000100	-0.003020	0.002860
Initial Guess C	0.000016	0.000100	-0.003020	0.002860
Initial Guess D	0.000016	0.000100	-0.003020	0.002860
Initial Guess E	0.000016	0.000100	-0.003020	0.002860

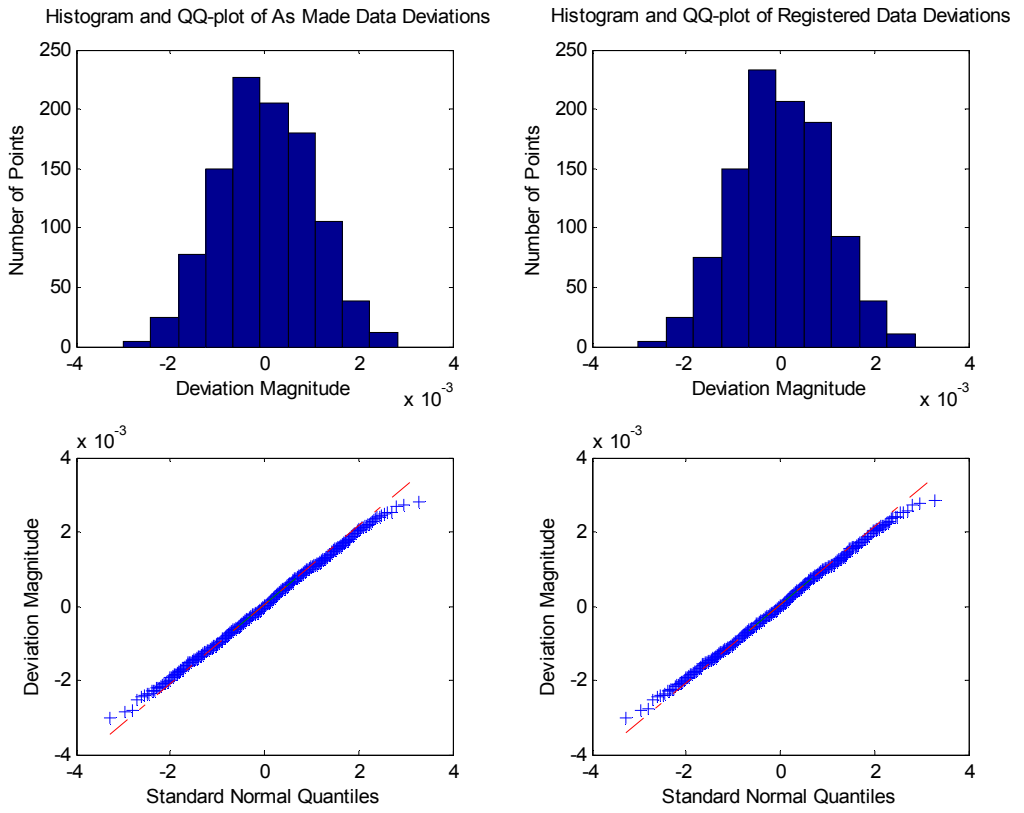

Figure 84: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for spherical geometry

Table 36: Deviation summary for 0.01 unit noise for spherical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	-0.000056	0.010070	-0.037348	0.029593
Initial Guess A	-0.000056	0.010049	-0.036502	0.029786
Initial Guess B	-0.000056	0.010049	-0.036502	0.029786
Initial Guess C	-0.000056	0.010049	-0.036502	0.029786
Initial Guess D	-0.000056	0.010049	-0.036502	0.029786
Initial Guess E	-0.000056	0.010049	-0.036502	0.029786

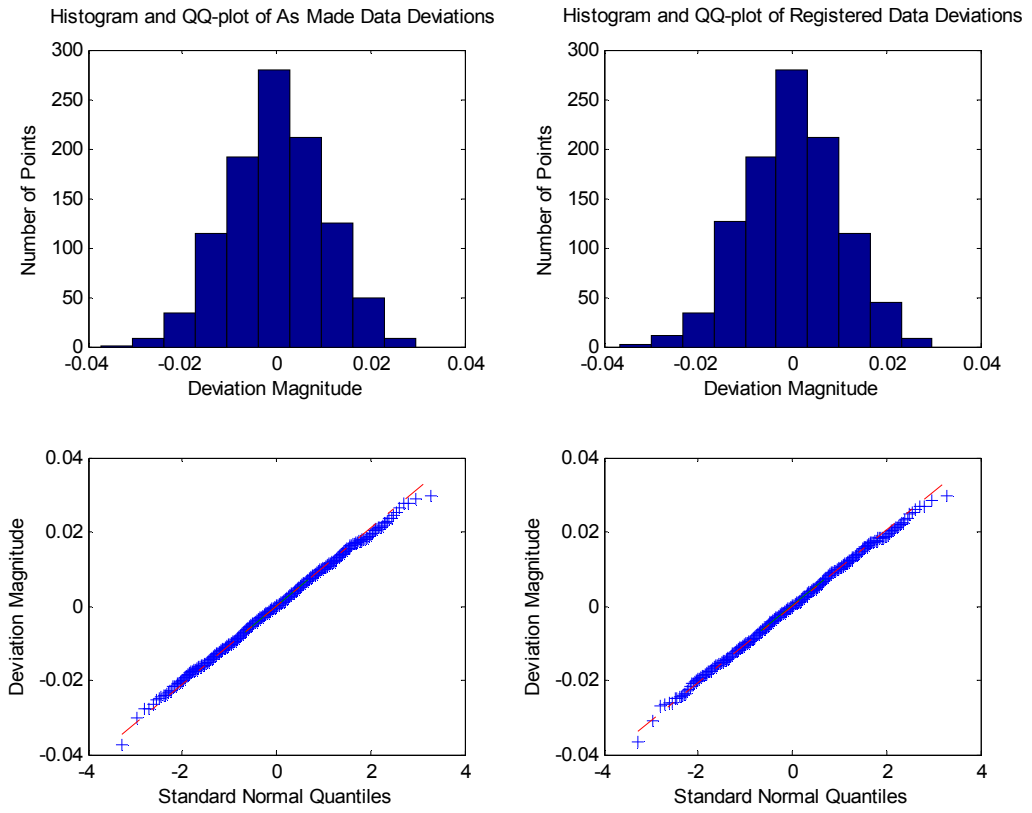

Figure 85: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for spherical geometry

Cylinder

In order to verify the registration process for cylindrical geometry, a CAD model containing a single cylinder is modeled. Again, a point cloud consisting of one thousand points is generated to lie exactly (to the precision of the machine) on the cylinder. The two planar faces representing the top and bottom of the cylinder are not used for any of the calculations in the registration process. For this type of geometry, the transformations that apply are x-translation, y-translation, x-rotation, and y-rotation. The other transformation types are excluded due to symmetry. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithm for cylindrical geometry. The procedure is similar to those previously discussed. Table 37 lists the starting positions of each point cloud before registration. Figure 86 graphically shows each transformed point cloud and the final registered location.

Table 37: Initial starting positions before registration for cylindrical geometry

	Tx (linear)	Ty (linear)	Rx (radians)	Ry (radians)
Optimal Fit	**0.0**	**0.0**	**0.0**	**0.0**
Initial Guess A	-0.01	0.02	-0.01	-0.02
Initial Guess B	0.45	0.5	0.3	0.2
Initial Guess C	0.3	-0.4	-0.2	-0.1
Initial Guess D	-0.25	0.3	0.35	0.4
Initial Guess E	-0.4	-0.25	0.4	0.3

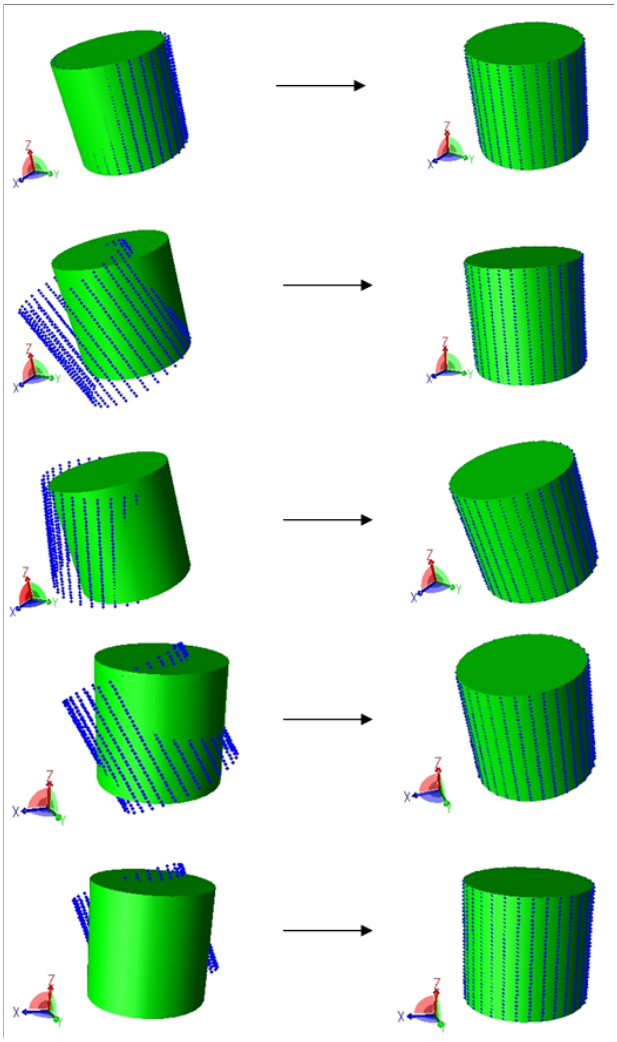

Figure 86: Initial guesses and registered point clouds for cylindrical geometry

Table 38: Deviation summary for test cases for cylindrical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	-0.000000	0.000000	-0.000000	0.000000
Initial Guess B	-0.000000	0.000000	-0.000000	0.000000
Initial Guess C	-0.000000	0.000000	-0.000000	0.000000
Initial Guess D	-0.000000	0.000000	-0.000000	0.000000
Initial Guess E	-0.000000	0.000000	-0.000000	0.000000

Table 39: Deviation summary for 0.001 unit noise for cylindrical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made	-0.000019	0.000962	-0.002815	0.003742
Initial Guess A	-0.000019	0.000957	-0.002838	0.003557
Initial Guess B	-0.000019	0.000957	-0.002838	0.003557
Initial Guess C	-0.000019	0.000957	-0.002838	0.003557
Initial Guess D	-0.000019	0.000957	-0.002838	0.003557
Initial Guess E	-0.000019	0.000957	-0.002838	0.003557

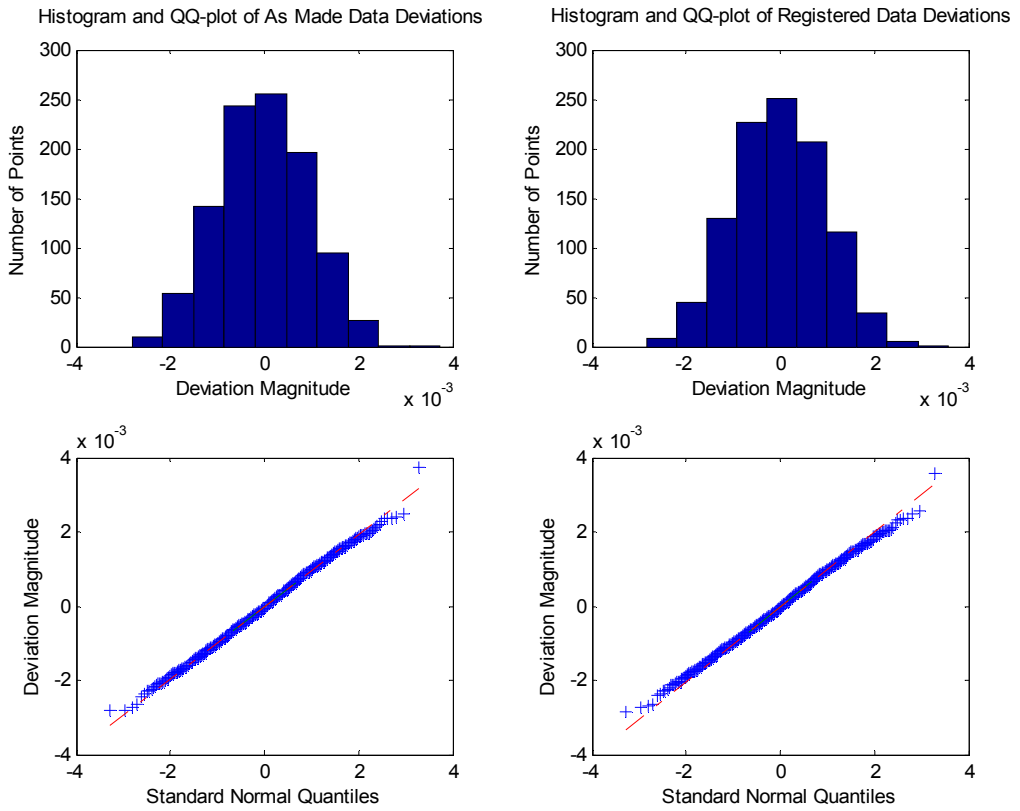

Figure 87: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for cylindrical geometry

Table 40: Deviation summary for 0.01 unit noise for cylindrical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000497	0.009822	-0.032913	0.029953
Initial Guess A	0.000497	0.009804	-0.033275	0.030596
Initial Guess B	0.000497	0.009804	-0.033275	0.030596
Initial Guess C	0.000497	0.009804	-0.033275	0.030596
Initial Guess D	0.000497	0.009804	-0.033275	0.030596
Initial Guess E	0.000497	0.009804	-0.033275	0.030596

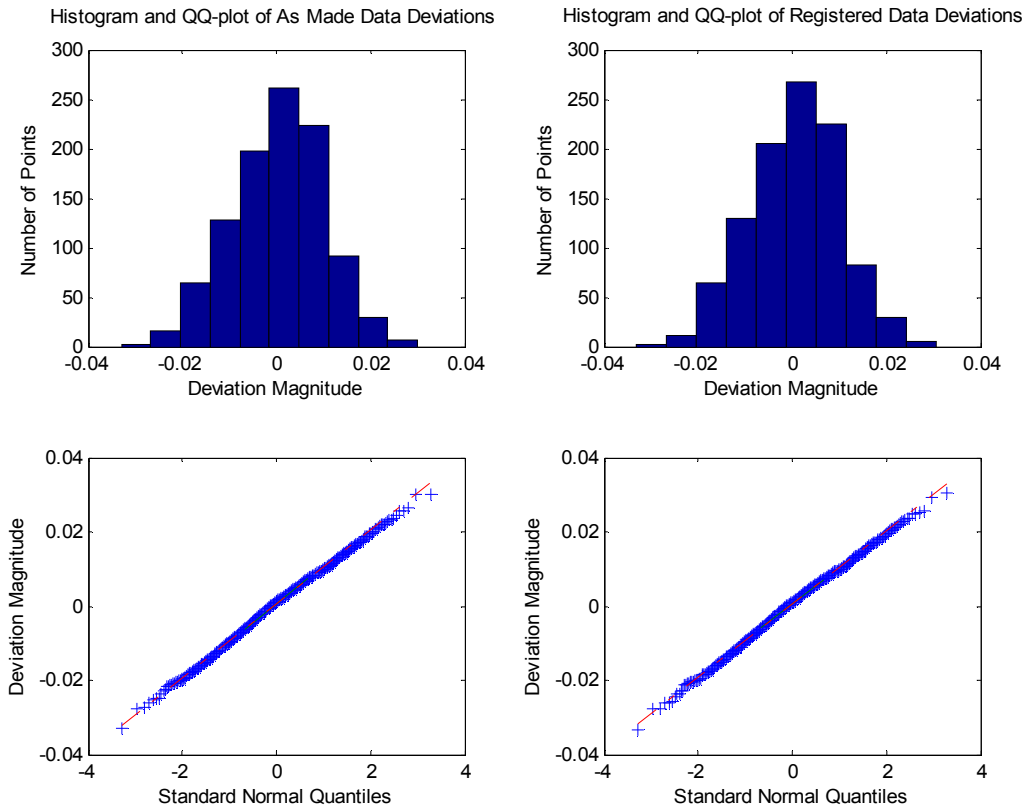

Figure 88: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for cylindrical geometry

Cone

In order to verify the registration process for conical geometry, a CAD model containing a single cone is modeled. A point cloud consisting of one thousand points is generated to lie exactly (to the precision of the machine) on the cone. The planar face of the cone is excluded from the analysis. All rigid body transforms except rotation about the axis of the cone are included. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithm for conical geometry. Table 41 lists the starting positions of each point cloud before registration. Figure 89 graphically shows each transformed point cloud and the final registered location.

Table 41: Deviation summary for test cases for conical geometry

	Tx (linear)	Ty (linear)	Tz (linear)	Rx (radians)	Ry (radians)
Optimal Fit	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
IG - A	-0.05	0.03	-0.02	-0.04	0.05
IG - B	0.45	0.4	0.5	0.4	0.2
IG - C	0.6	-0.5	-0.4	-0.2	-0.3
IG - D	-0.5	0.3	0.5	-0.25	0.4
IG - E	-0.25	-0.45	0.35	0.35	0.2

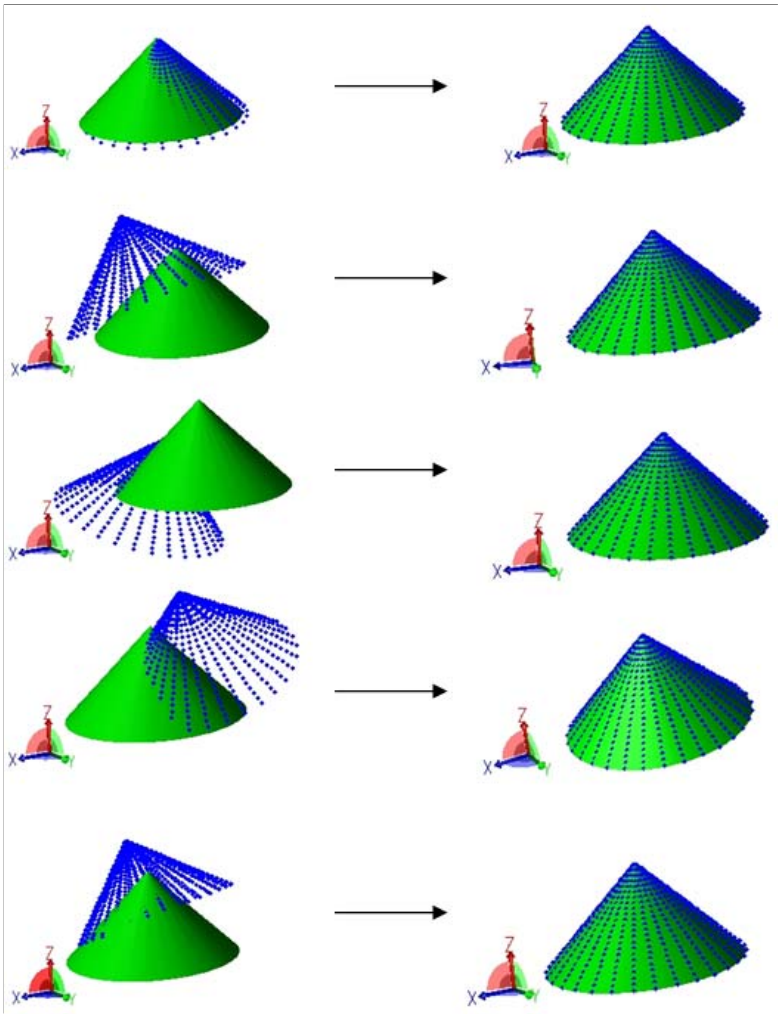

Figure 89: Initial guesses and registered point clouds for conical geometry

Table 42: Deviation summary for test cases for conical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	-0.000000	0.000000	-0.000000	0.000000
Initial Guess B	0.000000	0.000000	-0.000000	0.000000
Initial Guess C	-0.000000	0.000000	-0.000000	0.000000
Initial Guess D	-0.000000	0.000000	-0.000000	0.000000
Initial Guess E	-0.000000	0.000000	-0.000000	0.000000

Table 43: Deviation summary for 0.001 unit noise for conical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000012	0.001008	-0.003246	0.003182
Initial Guess A	0.000000	0.001001	-0.003197	0.003189
Initial Guess B	-0.000000	0.001001	-0.003197	0.003189
Initial Guess C	-0.000000	0.001001	-0.003197	0.003189
Initial Guess D	0.000000	0.001001	-0.003197	0.003189
Initial Guess E	-0.000000	0.001001	-0.003197	0.003189

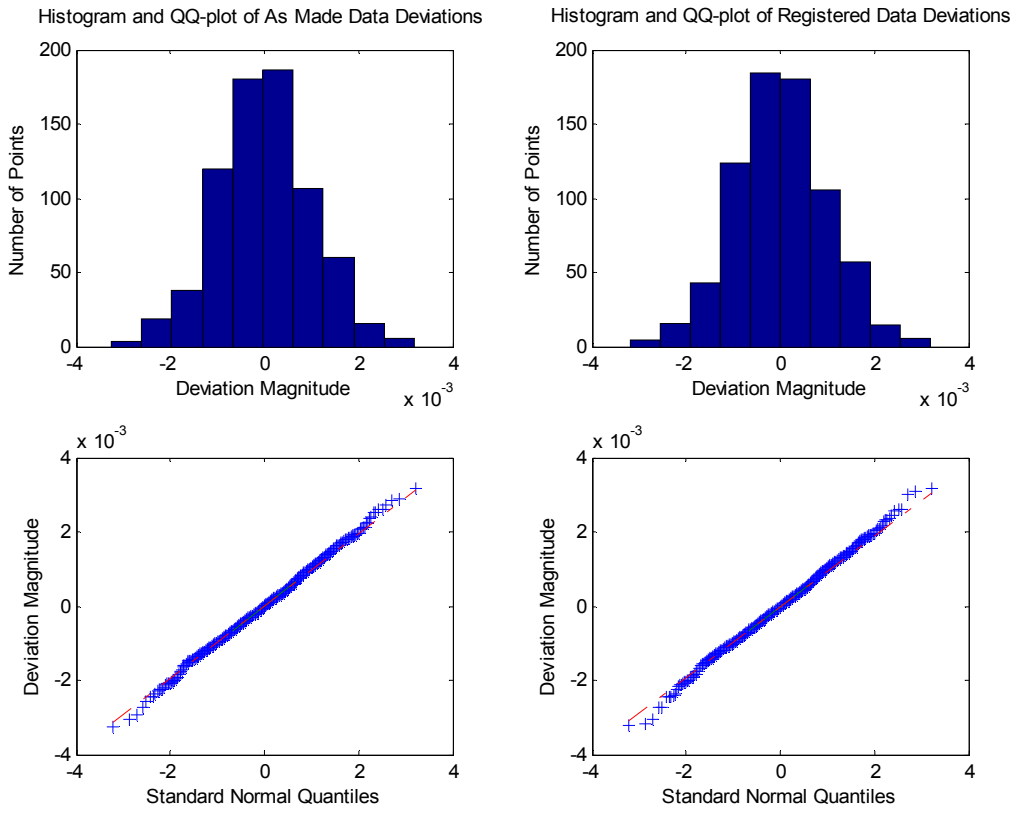

Figure 90: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for conical geometry

Table 44: Deviation summary for 0.01 unit noise for conical geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000118	0.009519	-0.027963	0.028804
Initial Guess A	0.000070	0.009506	-0.027727	0.029335
Initial Guess B	0.000070	0.009506	-0.027727	0.029335
Initial Guess C	0.000070	0.009506	-0.027727	0.029335
Initial Guess D	0.000070	0.009506	-0.027727	0.029335
Initial Guess E	0.000070	0.009506	-0.027727	0.029335

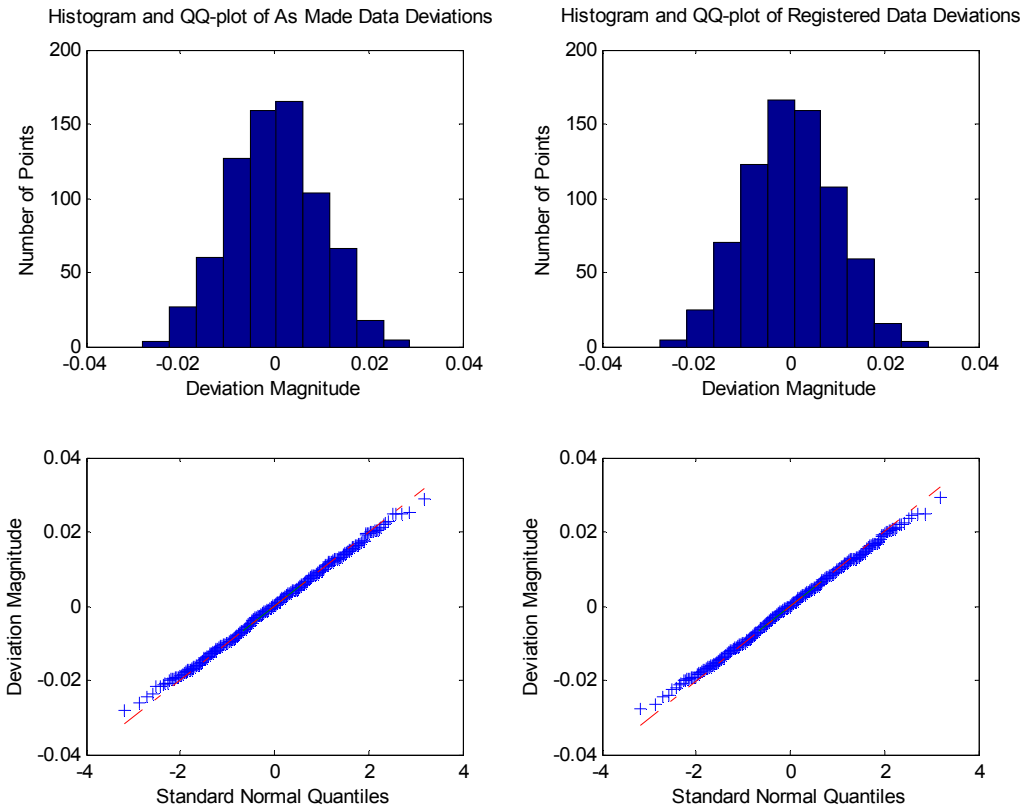

Figure 91: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for conical geometry

Torus

In order to verify the registration process for toroidal geometry, a CAD model of a torus is modeled. Again, a point cloud consisting of one thousand points is generated to lie exactly (to the precision of the machine) on the torus. For this type of geometry, the transformations that are applicable are x-translation, y-translation, z-translation, x-rotation, and y-rotation. Rotation about the z-axis is restricted due to the symmetry of the part. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithm for toroidal geometry. The procedure is similar to those previously discussed. Table 45 lists the starting positions of each point cloud before registration. Figure 92 graphically shows each transformed point cloud and the final registered location.

Table 45: Deviation summary for test cases for toroidal geometry

	Tx (linear)	Ty (linear)	Tz (linear)	Rx (radians)	Ry (radians)
Optimal Fit	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
IG - A	0.03	0.05	-0.03	-0.02	-0.04
IG - B	0.5	0.6	0.5	0.3	0.1
IG - C	0.3	-0.4	0.4	0.25	-0.4
IG - D	-0.6	0.5	-0.4	-0.3	0.3
IG - E	-0.4	-0.4	0.5	0.4	-0.25

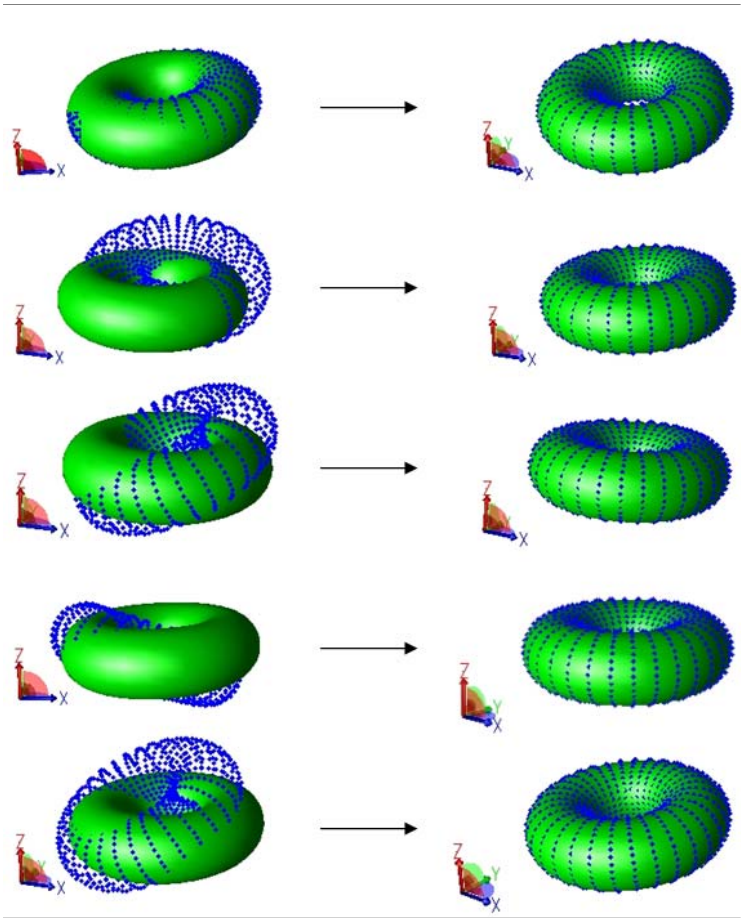

Figure 92: Initial guesses and registered point clouds for toroidal geometry

Table 46: Deviation summary for test cases for toroidal geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	-0.000000	0.000000	-0.000000	0.000000
Initial Guess B	-0.000000	0.000000	-0.000000	0.000000
Initial Guess C	-0.000000	0.000000	-0.000000	0.000000
Initial Guess D	-0.000000	0.000000	-0.000000	0.000000
Initial Guess E	0.000000	0.000000	-0.000000	0.000000

Table 47: Deviation summary for 0.001 unit noise for toroidal geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000041	0.000996	-0.003179	0.003061
Initial Guess A	0.000041	0.000995	-0.003141	0.003117
Initial Guess B	0.000041	0.000995	-0.003141	0.003117
Initial Guess C	0.000041	0.000995	-0.003141	0.003117
Initial Guess D	0.000041	0.000995	-0.003141	0.003117
Initial Guess E	0.000041	0.000995	-0.003141	0.003117

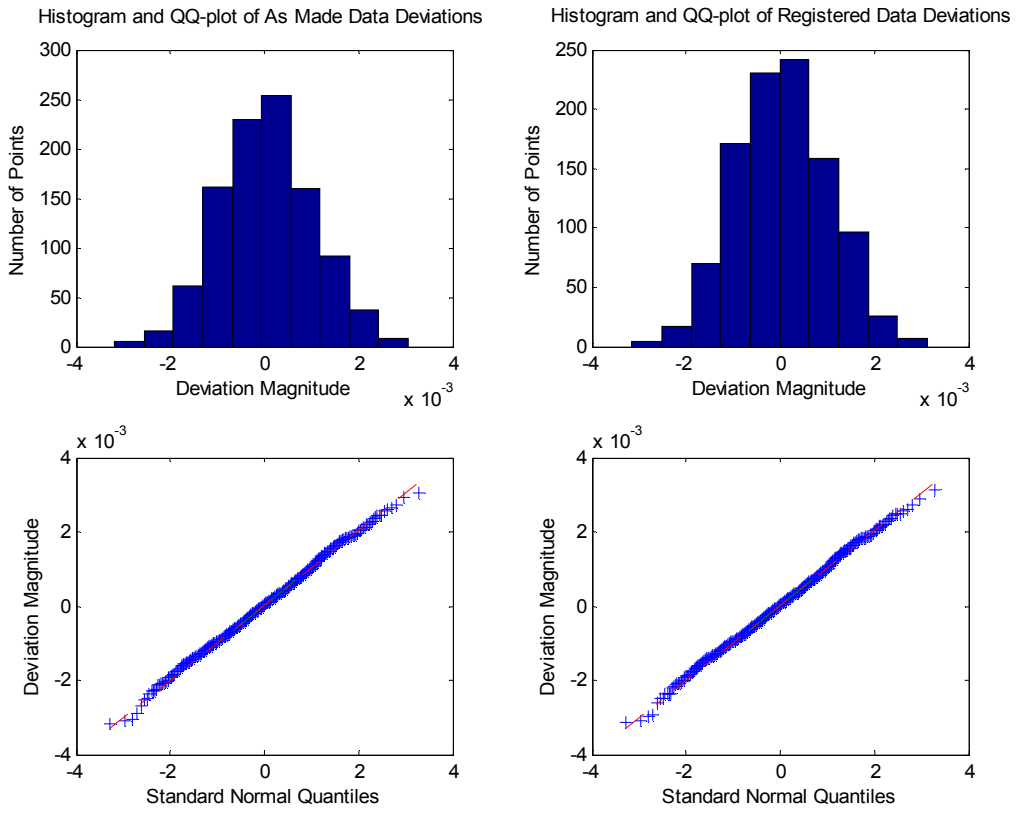

Figure 93: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for toroidal geometry

Table 48: Deviation summary for 0.01 unit noise for toroidal geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	0.000306	0.009695	-0.032786	0.029171
Initial Guess A	0.000307	0.009679	-0.032696	0.029098
Initial Guess B	0.000307	0.009679	-0.032696	0.029098
Initial Guess C	0.000307	0.009679	-0.032696	0.029098
Initial Guess D	0.000307	0.009679	-0.032696	0.029098
Initial Guess E	0.000307	0.009679	-0.032696	0.029098

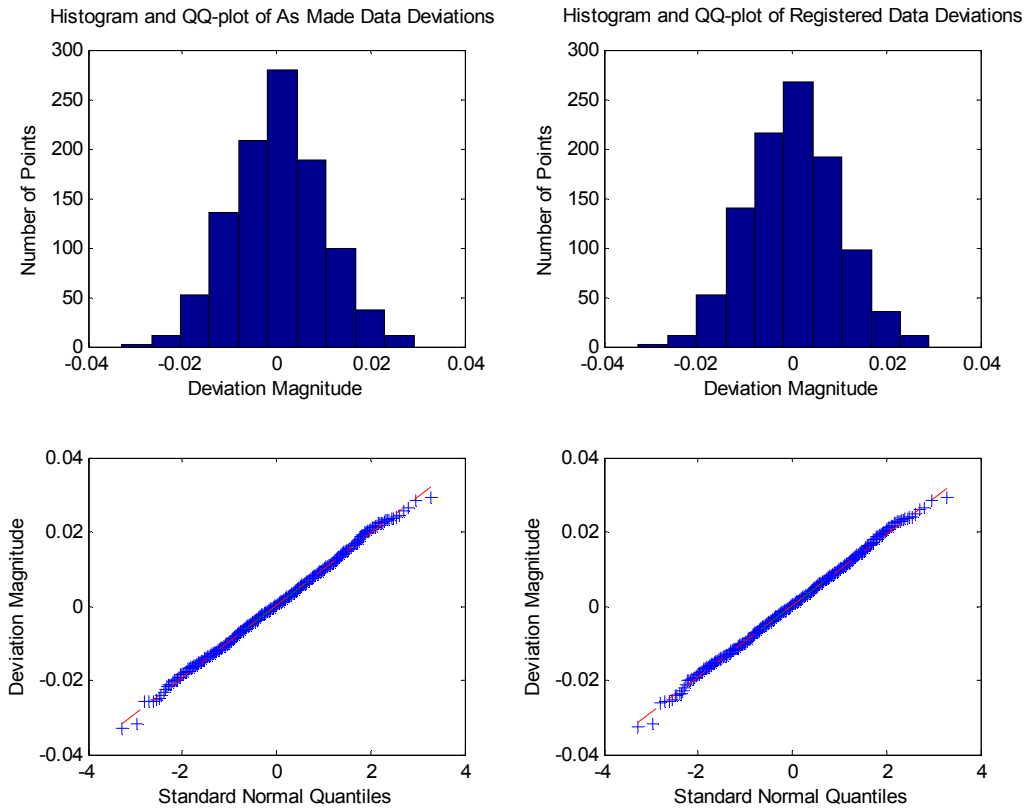

Figure 94: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for toroidal geometry

General Parametric Surface

In order to verify the registration process for general parametric geometry, a CAD model of a single spline surface is modeled. Again, a point cloud consisting of one thousand points was generated to lie exactly (to the precision of the machine) on the spline surface. For this type of geometry, all degrees of freedom are applicable. Five different initial starting points are devised in order to test the repeatability and robustness of the algorithm for general parametric geometry. The procedure is similar to those previously discussed. Table 49 lists the starting positions of each point cloud before registration. Figure 95 graphically shows each transformed point cloud and the final registered location.

Table 49: Deviation summary for test cases for parametric surface geometry

	Tx (linear)	Ty (linear)	Tz (linear)	Rx (radians)	Ry (radians)	Rz (radians)
Optimal Fit	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
IG - A	0.04	0.02	0.05	0.07	-0.05	0.04
IG - B	0.3	0.15	-0.25	-0.3	0.2	0.15
IG - C	0.2	-0.5	0.6	0.2	-0.35	0.3
IG - D	-0.5	0.5	-0.3	0.25	-0.3	0.35
IG - E	-0.4	-0.5	0.4	-0.4	-0.3	-0.4

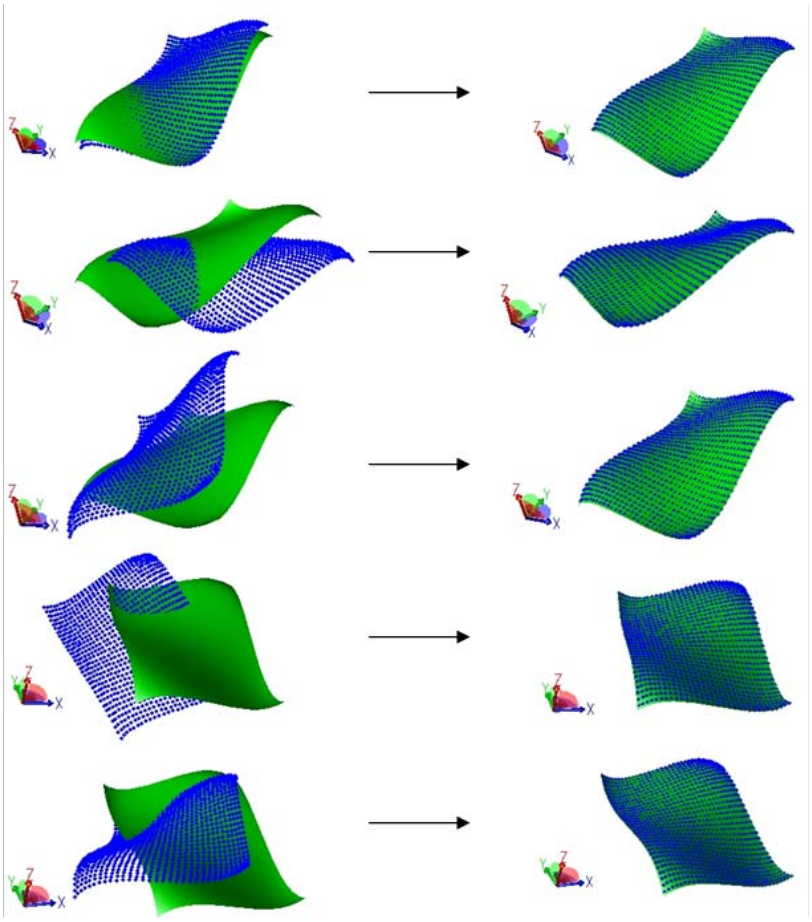

Figure 95: Initial guesses and registered point clouds for parametric surface geometry

Table 50: Deviation summary for test cases for parametric surface geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
Initial Guess A	0.000000	0.000000	-0.000000	0.000000
Initial Guess B	0.000000	0.000000	-0.000000	0.000000
Initial Guess C	0.000000	0.000000	-0.000000	0.000000
Initial Guess D	0.000000	0.000000	-0.000000	0.000000
Initial Guess E	0.000000	0.000000	-0.000000	0.000000

Table 51: Deviation summary for 0.001 unit noise for parametric surface geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	-0.000028	0.001048	-0.003701	0.002891
Initial Guess A	0.000001	0.001045	-0.003706	0.002952
Initial Guess B	0.000001	0.001045	-0.003706	0.002952
Initial Guess C	0.000001	0.001045	-0.003706	0.002952
Initial Guess D	0.000001	0.001045	-0.003706	0.002952
Initial Guess E	0.000001	0.001045	-0.003706	0.002952

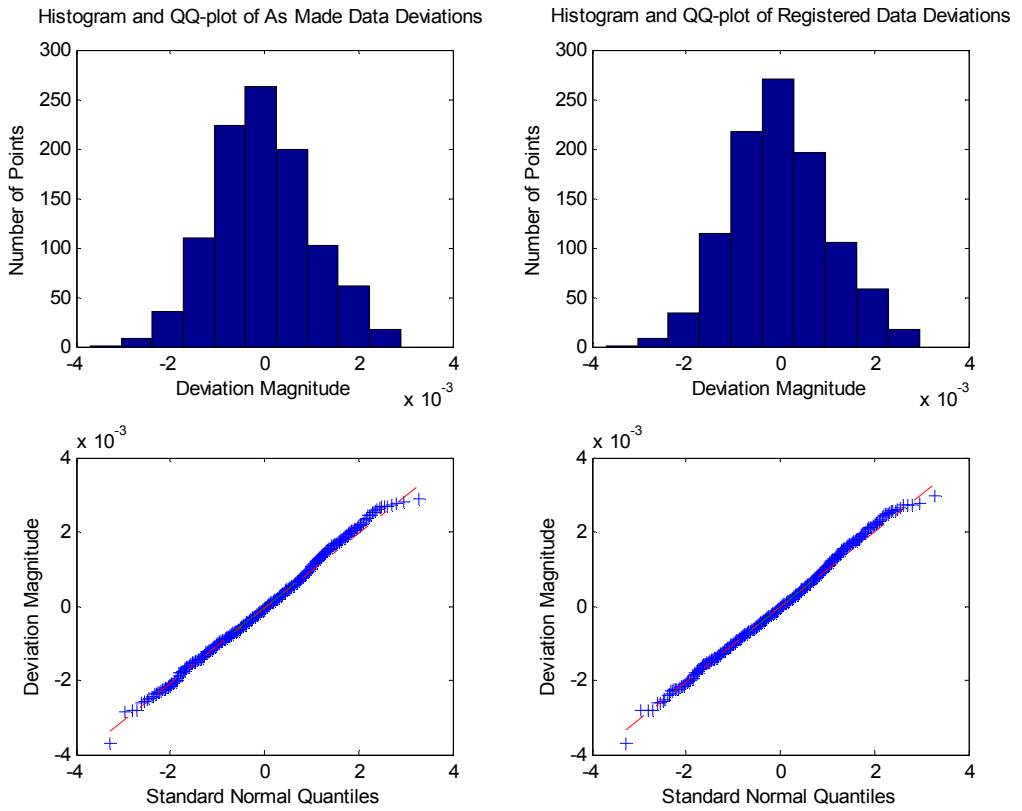

Figure 96: Histogram and QQ-plot of deviation vectors for 0.001 unit noise data for parametric surface geometry

Table 52: Deviation summary for 0.01 unit noise for parametric surface geometry

	Mean of errors	Std. Deviation	Min. Error	Max. Error
As-made Fit	-0.000156	0.009973	-0.030429	0.029659
Initial Guess A	0.000006	0.009945	-0.030649	0.030921
Initial Guess B	0.000006	0.009945	-0.030649	0.030921
Initial Guess C	0.000006	0.009945	-0.030649	0.030921
Initial Guess D	0.000006	0.009945	-0.030649	0.030921
Initial Guess E	0.000006	0.009945	-0.030649	0.030921

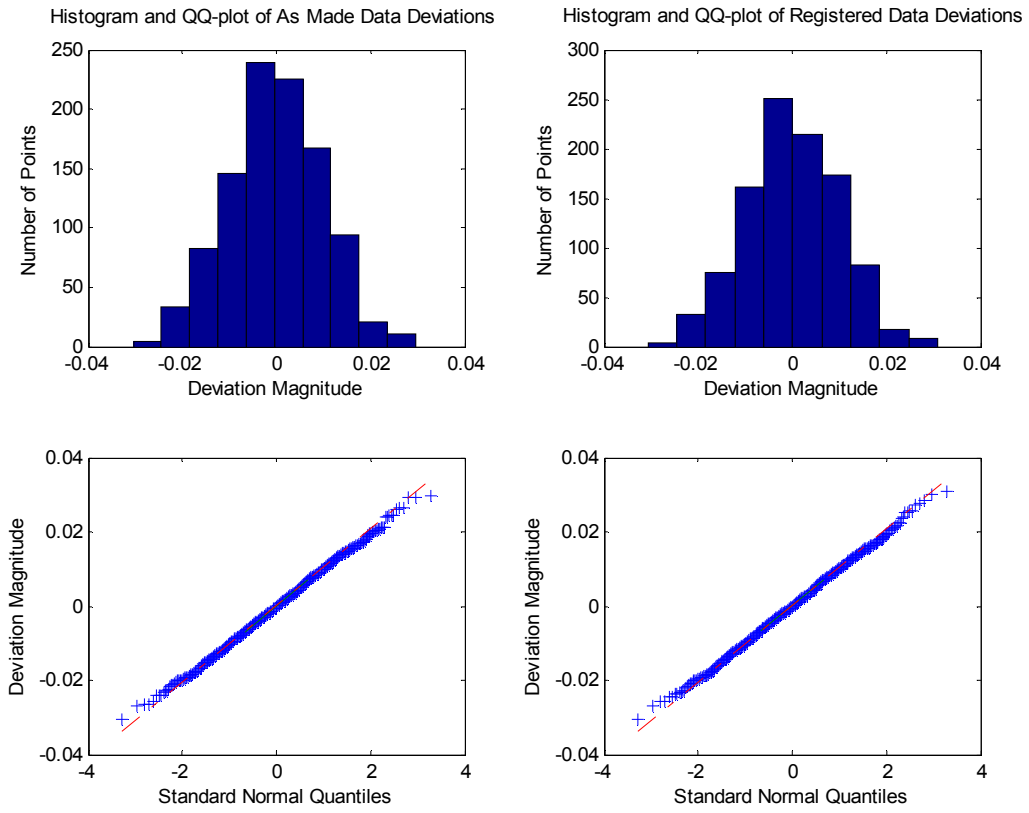

Figure 97: Histogram and QQ-plot of deviation vectors for 0.01 unit noise data for parametric surface geometry

REFERENCES

1. ANSI Y14.40.1M, "Graphical Symbols for Diagrams," American Society of Mechanical Engineers, 2002.

2. ANSI Y14.5M, "Dimensioning and Tolerancing," American Society of Mechanical Engineers, 1994.

3. Aracor, "Features of Aracor computed tomography systems," http://www.aracor.com/pages/products/ct.html, 2002.

4. B89.4.1, "Methods for Performance Evaluation of Coordinate Measuring Machines," American Society of Mechanical Engineers, 1997.

5. B89.4.10, "for Performance Evaluation of Coordinate Measuring System Software," American Society of Mechanical Engineers, 2000.

6. B89.7.2, "Dimensional Measurement Planning," American Society of Mechanical Engineers, 1999.

7. B89.7.3.1, "Guideline for Decision Rules: Considering Measurement Uncertainty in Determining Confirm to Specifications," American Society of Mechanical Engineers, 2002.

8. Bard Y., *Nonlinear Parameter Estimation,* Academic Press, New York, 1970.

9. Becker E., Ehrfeld W., Hagmann P., Mander A., Münchmeyer D., "Fabrication of microstructures with high aspect ratio and great structural heights by synchrotron radiation lithography, galvanoforming and plastic moulding (LIGA-process)," *Microelectronic Engineering*, vol. 4, pp. 35-56, 1986.

10. Besl P., McKay, N., "A method for registration of 3-D shapes", IEEE Transactions on Pattern Analysis and Machine Intelligence," vol. 14, no. 2, pp. 239-256, 1992.

11. Binnig G., Quate C., Gerber C., "Atomic force microscopy," *Physical Review Letters* 56**,** pp. 930–933, 1986.

12. Binnig G., Rohrer H., "Scanning tunneling microscopy," *IBM Journal of Research and Development*, v 44, n 1-2, pp. 279-93, Jan.-March 2000.

13. Bley P., Ehrfeld W., Schmidt D., "Effective separation of disparate mass mixtures in a single stage by double deflection separation nozzles," Univ of Tokyo Press, vol. 2, pp. 645-654, 1984.

14. Brodie I. and Muray J., *The Physics of Microfabrication,* Plenum Press, New York, 1982.

15. Ceremuga J., "Optimizing inspection of high aspect ratio microstructures using a programmable optical microscope," Master's Thesis, Georgia Institute of Technology, Atlanta, GA, 2003.

16. Choi, Woncheol, "Computational analysis of three dimensional measurement data," Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.

17. Claudet A., "Analysis of three dimensional measurement data and CAD models," Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA 2001.

18. Dennis J., Schnabel R., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Society for Industrial and Applied Mathematics, Philadelphia, 1996.

19. Fortune, S., "A sweepline algorithm for voronoi diagrams," *Algorithmica*, vol. 2, no. 2, pp. 153-174, 1987.

20. Griffith, J., Marchman H., Miller G., Hopkins L., "Dimensional metrology with scanning probe microscopes," *Journal of Vacuum Science & Technology*, vol. 13, no. 3, pp. 1100-1105, May-June 1995.

21. Griffiths S., Ting A., Hruby J., "The influence of mask substrate thickness on exposure and development times for the LIGA process," *Microsystem Technologies*, v 6, n 3, pp. 99-102, Feb. 2000.

22. Groot P., Deck L., "Surface profiling by frequency-domain analysis of white light interferograms," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2248, pp. 101-104, 1994.

23. Gunnarsson K., Prinz F., "CAD model based localization of parts in manufacturing," *IEEE Computer*, vol. 20, no. 8, pp.66-74, 1987.

24. Hall A., Degertekin L., "Integrated optical interferometric detection method for micromachined capacitive acoustic transducers," *Applied Physics Letters*, vol. 80, pp. 3859-3861, 2002.

25. Hartley F., Malek C., and Nguyen S., "X-Ray exposure of electron-deposited photoresist for conformal lithography on corrugated surfaces," *SPIE, Emerging Lithographic Technologies,* Santa Clara, California, USA, February 22, 1998.

26. Heller A., "Nondestructive characterization at the mesoscale," *Science and Technology Review*, Lawrence Livermore National Laboratory, October 2003.

27. Hibbard R., Bono M., “Meso-scale metrology tools: A survey of relevant tools and a discussion of their strengths and weaknesses,” *Proceedings of Machines and Processes for Micro-scale and Meso-scale Fabrication, Metrology, and Assembly*, pp. 70-72, 2003.

28. Hopp T., “Computational metrology,” *Manufacturing Review*, vol. 6, no. 4, pp. 295-304, 1993.

29. Hopp T., Levenson M., "Performance measures for geometric fitting in the NIST algorithm testing and evaluation program for coordinate measurement systems," *NIST Journal of Research*, September, 1995.

30. Hruby J., “LIGA technologies and applications,” MRS Bulletin, April 2001.

31. Kalpakjian S., Schmid S., *Manufacturing Processes for Engineering Materials 4th Ed.,* Prentice Hall, 2002.

32. Kim B., Razavi A., Degertekin L., Kurfess T., “Micromachined interferometer for MEMS metrology,” *Proceedings of Machines and Processes for Micro-scale and Meso-scale Fabrication, Metrology, and Assembly,* pp. 73-78, 2003.

33. Kim, B., Razavi, H. A., Degertekin, F. L., and Kurfess, T. R., “Micromachined interferometer for measuring dynamic response of microstructures,” *Proceedings of 2002 ASME International Mechanical Engineering Congress & Exposition*, MEMS Symposium, New Orleans, LA, November 2002.

34. Larrabee R., Postek M., “Parameters characterizing the measurement of a critical dimension,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. CR52, pp. 2-24, 1994.

35. Levenberg K., “A method for the solution of certain problems in least squares,” *Quarterly Applied Mathematics*, no. 2, pp. 164-168, 1944.

36. Madou M., *Fundamentals of Microfabrication*, CRC Press, Boca Raton, FL, 1997.

37. Marchman H., Griffith J., Guo J., Frackoviak J., Celler G., “Nanometer-scale dimensional metrology for advanced lithography,” *Journal of Vacuum Science & Technology*, vol. 12, no. 6, pp. 3585-3590, Nov.-Dec. 1994.

38. Marquardt D., “An algorithm for least squares estimation of nonlinear parameters,” *SIAM Journal of Applied Mathematics*, no. 11, pp. 431-441, 1963.

39. Moré J., “The Levenberg-Marquardt algorithm: Implementation and theory,” *Numerical Analysis,* Lecture Notes in Math, pp. 105-116, 1977.

40. Moré J., “User guide for MINPACK-1,” *Argonne National Labs Report*, ANL-80-74, 1980.

41. Mortenson M., *Geometric Modeling, 2nd Ed.,* John Wiley and Sons, Inc., New York, 1997.

42. O'Mahony C., Hill M., Brunet M., Duane R., and Mathewson A., "Characterization of micromechanical structures using white-light interferometry," *Measurement Science and Technology*, vol. 14, pp. 1807-1814, 2003.

43. Peggs G., Lewis A., and Leach R., "Measuring in three dimensions at the mesoscopic scale," *Proceedings of Machines and Processes for Micro-scale and Meso-scale Fabrication, Metrology, and Assembly*, pp. 53-57, 2003.

44. Peggs, G., Lewis A., Oldfield S., "Design for a compact high-accuracy CMM," *CIRP Annals - Manufacturing Technology*, vol. 48, no. 1, pp. 417-420, 1999.

45. Postek M., "Scanning electron microscope metrology," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. CR52, pp. 46-90, 1994.

46. Prasad S., Hall A., Dugger M., "Characterization of sidewall and planar surfaces of electroformed LIGA parts," Sandia National Laboratories, August 2000.

47. Roth J., Felkel E., Groot P., "Optical metrology," *Proceedings of Machines and Processes for Micro-scale and Meso-scale Fabrication, Metrology, and Assembly*, pp. 87-92, 2003.

48. Rusinkiewicz S., Levoy M., "Efficient variants of the ICP algorithm," *Third International Conference on 3D Digital Imaging and Modeling*, 2001.

49. Sahoo K., Menq C., "Localization of 3D objects having complex sculptured surfaces using tactile sensing and surface description," *Journal of Engineering for Industry*, vol. 113, pp. 85-92, 1991.

50. Schellekens, P., Haitjema H., Pril, W., "A silicon-etched probe for 3-D coordinate measurements with an uncertainty below 0.1 μm," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 6, pp. 1519-1523, December 2001.

51. Schenk R., Halle O., Muellen K., Ehrfeld W., Schmidt M., "Highly sensitive resist material for deep x-ray lithography," *Microelectronic Engineering*, vol. 35, no. 1-4, pp. 105-108, Feb. 1997.

52. Shakarji C., "Least squares fitting algorithms of the NIST algorithm testing service," *Journal of Research of the National Institute of Standards and Technology*, vol. 103, no. 6, pp. 633-641, 1998.

53. Sheats J., *Microlithography: Science and Technology*, Marcel Dekker, Inc., New York, 1998.

54. Spatial Technologies, "ACIS R10 help documentation," Spatial Technologies, Westminster, CO, 2002.

55. St. Clair L., Mirza A., "Metrology for MEMS manufacturing," *Sensors Magazine*, vol. 17, no. 7, July, 2000.

56. Tucker T., "A new method for parametric surface registration," Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, 2000.

57. Veeco, "Scanning probe/atomic force microscopy: Technology overview and update," Veeco Instruments, Inc., Santa Barbara, CA, 2003.

58. VIEW Engineering, "Voyager 6x12 specifications," http://www.vieweng.com, 2003.

59. Wyant J., "White light interferometry," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4737, pp. 98-107, 2002.

60. Zworykin V., Hillier J., Snyder R., "A scanning electron microscope," *ASTM Bull.* vol. 117, pp.15-23, 1942.

61. Zygo Corporation, "NewView 5000 specifications," Zygo Corporation, Middlefield, CT, 2003.